

LINUX YAZ KAMPI 2016 by MEHMET KILIÇ

GNU-LINUX DEBİAN TERMİNAL KOMUTLARI 1

Bir Gnu/Linux işletim sisteminde sistem ve kullanıcı dizinleri kök dizini altında bulunur.

Kök dizin altındaki temel klasörler

Linux Dosya Sistemlerde (kısaca FSSTND) standartına göre kök dizin altında bulunan dizinlerin görevleri özetle şöyledir:

/bin : Olması zorunlu temel komut dosyalarını içerir.
/boot : Başlangıç için gerekli dosyaları bulundurur.
/dev : Donanım dosyaları vardır.
/etc : Sistem ayarlarını barındırır.
/lib : Kütüphane dosyaları ve çekirdek modülleri bulunur.
/media : Kaldırılabilir aygıtların (CD-ROM, USB bellek vb.) sisteme eklendiği klasördür.
/mnt : Sistem açılışında otomatik olarak bağlanan sabit disk bölümleri bu dizin altında eklenir.
/opt : Üçüncü parti kullanıcı programlarının kurulması içindir.
/sbin : Sistemi yöneticisiyle ilgili çalıştırılabilir dosyaları tutar.
/srv : Sistemin sunduğu hizmetlerle alakalıdır.
/tmp : Geçici dosyaları tutmak içindir.
/usr : Tüm kullanıcılarca paylaşılan verileri içeren dizindir.
/var : Log dosyaları, e-posta ve yazıcı kuyrukları gibi değişken verileri barındırır.

/bin

İşletim sisteminizi kullanmak için gereken birçok komut /bin klasörü altındadır. cat, mkdir, cp, ls, mv, rm vb. temel komutların hepsi buradadır. Sistem boot ettiğinde, ilk olarak /bin klasörü çalışır hâle getirilir. Sistemde ne kadar ciddi bir sorun olursa olsun /bin klasöründeki komutlar çalışmaya devam eder. Sisteminizde bir sorun meydana geldiğinde /bin klasörü altındaki komutları kullanarak sistemi onarabiliriz.

/boot

Boot, işletim sisteminin yüklenme evresidir. /boot klasörü, boot işlemi için gerekli olan tüm dosyaları içerir (çekirdek görüntüsü, sistem haritası, önyükleyici yapılandırması gibi). Bilgisayarın başlangıç (boot) aşamasında gerekmeyen ayar ve yapılandırma dosyaları burada bulunmaz; başka klasörlerden gerektiği zamanlarda yüklenir.

/dev

Linux'ta her şey bir dosyadır; donanım aygıtları da öyle. USB girişleri, seri ve paralel portlar, diskleriniz, CD-ROM'larınız vb... Bütün aygıtlar /dev klasörü altında tutulan dosyalardan ibarettir. Örneğin /dev altında bulunan hda1 dosyası, sabit diskinizi temsil eder. Ya da /dev/dsp, ses aygıtınızdır. Bunları programlar vasıtasıyla kullanırsınız; ancak bu dosyalar üzerinden doğrudan müdahale etmek de mümkündür. Mesela "cat /boot/vmlinuz > /dev/dsp" yazarak Çekirdeğin sesini duyabilirsiniz.

Bazı önemli aygıt dosyaları:

```
/dev/ttyS0 : Fare, modem gibi aygıtların bağlandığı seri port (COM 1).  
/dev/psaux : PS/2 girişi; fare ve klavyeler içindir.  
/dev/lp0 : Paralel port (LPT 1); yazıcı, tarayıcı vs...  
/dev/dsp : Birincil ses aygıtı  
/dev/usb : USB aygıtları  
/dev/sda : SCSI aygıtlar, USB bellekler, harici CD-ROM'lar vs...  
/dev/scd : SCSI CD-ROM'lar  
/dev/js0 : Oyun çubuğu (Joystick)
```

/etc

İşletim sistemini bir vücuda benzetirsek, /etc klasörünü sinir sisteminin merkezi olarak görebiliriz. Sisteme dair bütün ayarları barındırır, bulunduğu bilgisayara özel birçok yapılandırma bilgisini içerir. Durağandır, çalıştırılmak için değildir. /etc klasörü içerisinde bulunan bütün klasör ve yapılandırma dosyalarını tek tek incelememiz mümkün değil. Ancak son kullanıcının işine yarayacak bir ipucu verebiliriz.

Sürekli DNS problemleri yaşayanlar, /etc/resolv.conf dosyasını root (yetkili) olarak açıp yeni DNS adresleri ekleyebilir.

/home

home klasörü kullanıcıların kalesi olarak tabir edilir. home klasörü içerisinde her kullanıcının kendi adında bir alt klasörü bulunur. (örneğin /home/ahmet , /home/ayşe gibi). Kullanıcıların kişisel verileri, kullandığı programlarda yaptığı ayar değişiklikleri, yapılandırmaları tutulmaktadır. Kullanıcının çeşitli programlarda yaptığı ayarları barındıran dosyalar gizli dosya oldukları için görüntülenebilmeleri için gizli dosyaların görünür hale getirilmesi gereklidir. Kullanıcılar /home dizinini elbette belge, müzik, resim vb. dosyaları düzenli, tertipli bir şekilde depolamak amacıyla da kullanabilirler. /home dizini Windows'taki kullanıcı dizinlerine göre çok daha güvenli bir yapıya sahiptir. Çünkü Linux'ta bir başkasının ev klasörüne müdahale edemezken, Windows'ta çok zorlanmadan istediğinizi yapabilirsiniz.

/home dizini, aynı zamanda kullanıcı ayar dosyalarını barındırıyor olması nedeniyle bu dizinini, Windows'taki **Documents and Settings** ya da **Application** klasörlerine benzetebiliriz.

/initrd

initrd, "initial ramdisk" kısaltmasıdır. Anlamı, yaklaşık olarak "Başlangıç Bellek Diski" şeklinde ifade edilebilir. Boot aşamasında ilk önce çekirdek (kernel) yüklenir. Bundan sonra bilgisayarınızın belleğinde bir Bellek Diski oluşturulur. Oluşturulan Bellek Disk üzerinde / (root) yansı açılır ve kök dizin olarak monte edilir. /initrd bu işlemlerin yapılması ve Linux'un yüklenmesi için gereklidir.

/lib

Çekirdek modülleri ve paylaşılan kütüphane dosyaları bu klasörde bulunur. Var olan çekirdek modüllerini /lib/modules/[versiyon_numarası] içerisinde bulabilirsiniz. Bahsedilen kütüphane dosyalarıysa, sistemi başlatmak ve /bin ile /sbin içerisindeki komutları çalıştırmak için gereklidir. Paylaşılan kütüphane dosyalarını, Windows'ta DLL ile eş tutabiliriz. Linux'ta kütüphane dosyalarının sonu ".so" ile biter.

/lost+found

İngilizce bir terim olan “Lost and Found” kayıp eşya bürosu demektir. /lost+found klasörü de tam olarak bu işlevi görmektedir. Bazen sisteminizde herhangi bir problem olur; örneğin bilgisayarı resetlerseniz, elektrik gider sonrasında bilgisayarı yeniden başlatırsınız. Bu gibi durumlarda Linux'ta fsck (File System Check) komutu devreye sokulur. Bu komut Windows'taki Scandisk programına benzetilebilir. Düzeltilemeyen bir sorun varsa, bağlantıları kopmuş kayıp dosyalar ortaya çıkmışsa, bunlar /lost+found altına atılır. Kısaca özetlersek; kötü bir sistem kapanmasından sonra, olması gereken bazı dosyaları bulamıyorsanız, kayıp eşya bürosuna bakmanızda yarar var. Ancak bu klasöre girmek istediğinizde erişimi engelleyen bir ileti ile karşılaşılırsınız, bu klasörün içeriğine ulaşabilmek için [dosya yöneticisini](#) tam yetki ile açmanız gereklidir. Bunun için `sudo -H nautilus` komutunu kullanmalısınız.

/media

CD-ROM, disket sürücü, flash bellek gibi çıkarılabilir aygıtlar buraya bağlanır. En basit tanımla, çıkarılabilir aygıtların, bağlantı noktası (mount point) olarak düşünülebilirsiniz.

/mnt

/media klasörünün aksine çıkarılabilir aygıtlar yerine, sistem açılışında otomatik olarak bağlanan sabit disk bölümleri ve donanım aygıtlarının bağlanması içindir.

Bağlama (mount) işlemi, herhangi bir depolama ortamını, işletim sisteminin kullanmasını sağlar. Nereye bağladığınız sizin tercihinizdir, değiştirmeniz mümkündür. Yani bir diski, /media veya /mnt klasörüne ya da bir başka yere bağlamanız fark etmeyecektir. /media ve /mnt genel kabul görmüş bağlantı noktalarıdır.

/opt

İşletim sisteminden bağımsız, sistem için zorunlu olmayan 3. parti kullanıcı programları bu dizinde bulunur. Örneğin; google earth programını indirip kurmak istediğinizde, 'default' olarak kurulacağı nokta, /opt/google-earth adresidir.

Elbette üçüncü parti bir programı kurarken bu kurulum konumunu değiştirebilir, size uygun gelen bir başka konuma yükleyebilirsiniz. Ancak daha önce de bahsettiğimiz gibi bazı şeyler genel kabule dayanır.

/proc

Süreçler, sistem belleği, bağlı aygıtlar, donanım yapılandırmalarıyla ilgili bilgileri içeren özel bir “sanal” dosya sistemidir. Bildiğimiz anlamda fiziksel dosyalar bulundurmaz; sistem durumuna dair bilgi içeren sanal dosyaları vardır. Bir bilgi alma merkezi olarak görülebilir, birçok uygulama buradaki bilgilerden yararlanmaktadır. Örneğin "cat /proc/swaps" yazarak sisteminizdeki takas dosyalarına dair bilgi alabilir ya da "cat /proc/cpuinfo" komutuyla işlemcinizin özelliklerini görebilirsiniz.

/proc klasörü içersindeki dosyalar, sadece sistem durumunu görüntülemek için kullanılmaz, gerektiğinde sistemde ayarlama yapmak için de kullanılabilir. Fakat son kullanıcılara hitap eden bir konu değildir.

/root

Linux/Unix sistemlerde, işletim sistemine her türlü müdahalede bulunabilme yetkisine sahip, "root" adıyla tanımlanmış, süper yetkili özel bir kullanıcı hesabı vardır. /root dizini, bu özel kullanıcı hesabının ev dizinidir. Root kullanıcısına "kök kullanıcı" da denilir.

Kullanıcıların, sistemi root hesabıyla açma ihtiyacı bulunmaması nedeniyle ve ayrıca sistemi root olarak açmanın güvenlik zaafiyetine yol açabilecek olması nedeniyle pek çok Linux dağıtımında root hesabıyla sisteme giriş yapılması, öntanımlı olarak engellemiştir. Linux dağıtımlarında, yetki gerektiren bir işlemin yapılabilmesi için sistem root olarak açılmaz, bunun yerine geçici olarak root hakları elde edilir. Bunun için, önce bir uçbirim komut satırı penceresi açılır. Sonra kullanmakta olduğunuz Linux dağıtımına bağlı olarak su, su - , su root ya da sudo komutlarından biri girilir ve ardından root kullanıcısının (ya da yönetici hesabın) parolası girilir.

/sbin

Linux'ta normal kullanıcının kullanabileceği komutlarla, kök kullanıcının (root) kullanabileceği komutlar ayrılmıştır. root tarafından kullanılacak bakım ve yönetim için kullanılan önemli programlar, /sbin altında tutulur. Daha az öneme sahip yönetim komutlarıysa, /usr/sbin klasöründedir. Eğer yerelde, yani kullandığınız makineye özgü kök kullanıcı (root) komutları bulunuyorsa, bunları da /usr/local/sbin altında bulabilirsiniz.

/usr

Unix ilk çıktığında, kullanıcılara ait ev klasörleri, /usr altında tutulurdu. Örneğin; "cagatay" isimli bir kullanıcının ev klasörü /usr/cagatay şeklindeydi. Bu yöntem zamanla değişti ve /home klasörü doğmuş oldu. /usr klasörü hâlâ çok önemli bir yapıdır ve işletim sisteminizde kullandığınız her şeyle ilişkisi bulunur.

Kurduğunuz pek çok program /usr içine kurulur. Her ne kadar Linux'ta kurulan bir programa ait tüm dosyalar usr dizini içine atılmıyor olsa da usr dizini Windows'taki Program Files klasörü ile benzer görev görmektedir. /opt adresi işletim sistemi dışında gelen 3.parti programlar içindir. Fakat işletim sistemi aracılığıyla ya da paket yönetim sistemlerini kullanarak yüklediğiniz her şey /usr altına aktarılır. Aynı zamanda yüklediğiniz programların çalışmak için ihtiyaç duyacağı kütüphane dosyaları, /usr/lib altındadır.

/usr, daha geniş bir tanımla; tüm kullanıcılarca paylaşılan verileri (programlar, komutlar, kütüphaneler, dokümanlar gibi) içeren dizindir. /usr ile ilgili söylenebilecek bir başka nokta da "local" klasörüdür. Linux, sunucu olarak birçok istemciye hizmet verebilecek bir işletim sistemidir. Bir Linux sunucu (server) kurarsanız, ona bağlanan yüzlerce istemci (client) olabilir. Her istemci bilgisayara, /usr altındaki programların ayrı ayrı yüklenmesi gerekmez; bir başka konumdan bu komutları çalıştırabilirsiniz. Ancak /usr altında bulunan "local" klasörü sadece kullandığınız makineye özeldir. Örneğin /usr/local/bin klasörü içinde bulunan bütün komutlar, direkt olarak kullandığınız makineye yüklenmiştir. /usr/bin komutları ise geneldir ve bir ağ üstündeki bütün makinelerden erişilebilir. "local" gördüğünüzde, bunun sadece sizin makinenize özel olduğunu bilin.

/var

Log dosyaları, e-posta ve yazıcı kuyukları gibi değişken sistem bilgilerini barındırır. Sisteminize dair tutulan log'ları buradan görebilir; güvenlik durumunu buradan kontrol edebilirsiniz.

/tmp

Geçici dosyalar içindir. Birçok program, burayı geçici depolama alanı olarak kullanır. /tmp klasörünün içeriği genellikle KB'lar mertebesinde kalır ve genellikle işletim sistemi yeniden başlarken içindeki dosyalar silinir. Her ne kadar tmp klasörü geçici dosyalar için de olsa bu klasör altında bulunan dosyaları, ne yaptığınızdan emin değilseniz, **kesinlikle silmemelisiniz!** Aksi takdirde sisteminizde ya da bazı programlarda sorun çıkabilir. Örneğin açık bir socket dosyasını sildiğinizde, onu kullanan programa ve kendinize sorun çıkartabilirsiniz.

KOMUT OPSİYON PARAMETRE

~ → /home/username demektir

/ → root yani kök olarak bilinir. Sistemin en üst seviyesidir. İçide home dizini olmak üzere sistem dizinleri ve dosyaları bulunur.

su → super user yani root olarak sisteme erişmemizi sağlar burada giriş yapmak için root parolanızı girmeniz gerekir

sudo su → root olarak sisteme erişmemizi sağlar. Burada kullanıcı parolanızı girmeniz gerekir

cd gidilmekistenenyol → cd komutundan sonra gidilmek istenen yol girilirse o yola girer

ls → bulunduğu dizini listeler

ls -l → bulunduğu dizini dosya indeksleri ile listeler

ls -al → bulunduğu dizindeki her türlü (gizli dosyalar vs.) dosyayı indeksleri ile listeler

ctrl + shift + t → yeni terminal sekmesi açar

ctrl + shift + w → yeni terminal sekmesini kapatır

ctrl + shift + c → terminal üzerinde kopyalama işlemi yapar

ctrl + shift + v → terminal üzerinde yapıştırma işlemi yapar

ctrl + a → imceli satır başına getirir

mkdir dizinadı → bulunduğu yola girilen dizin adında yeni bir dizin oluşturur

rm -r dizinadı → bulunduğu yerde girilen dizini siler

touch dosyaadı → girilen dosya adında bir binary dosyası oluşturur (metin dosyası vs)

rm dosyaadı → girilen dosya adındaki binary dosyasını siler

. → bulunulan dizini temsil eder

.. → bulunulan dizinin bir üst dizini temsil eder

ps aux → o anki çalışan proseslerin durumunu gösterir(PID numarası ile birlikte)

ps aux | grep programadı → girilen program adına ait çalışmayı PID numarasıyla birlikte gösterir

killall programdı → girilen programı kapatır

kill -9 programınPIDnumarası → girilen PID numarasına sahip programı kapatır

cp kopyalanacakdosyanınyolu kopyalanacağı yol → kopyalacak dosyanın yolu belirtilen dosyayı kopyalanacağı yol belirtilen yere kopyalar

cp kopyalanacakdizininyolu kopyalanacağı yol → kopyalacak dizinin yolu belirtilen dizini kopyalanacağı yol belirtilen yere kopyalar

mv taşıncakdosyanınyolu taşınacağı yol → taşınacak dosyanın yolu belirtilen dosyayı taşınacağı yol belirtilen yere taşır

mv -r taşıncakdizininyolu taşınacağı yol → taşınacak dizinin yolu belirtilen dizini taşınacağı yol belirtilen yere taşır

dd if=/yazdırılacakdosya(iso vb) of=/dosyanıyazdırılacağıyerinyolu → yazdırılacak (iso dosyası vb) dosyaları yolu belirtilen (usb vb) aygıtlara yazdırmaya yarar

sudo ls -l /dev/disk/by-id/*usb* → içinde usb geçen dosyaları listeler

top → çalışan prosessleri gösterir

top -d süre → girilen sürede güncellenen prosessleri gösterir

top -u aranacakuygulama → girilen uygulamaya ait prosessleri gösterir

çalışan proseslerde görünen **nice** → çalışma önceliği sırasını belirler . Nice değerleri -20 ile +20 arasındadır. -20 ise en öncelikli programı belirtir.

top -P PIDnumarası → girilen PID numarasına ait uygulamanın prosesslerini gösterir

pgrep programadı → girilen programın PID sini öğrenmemizi sağlar

preg -l programadı → girilen programı ismi ve PID si ile birlikte gösterir (aynı isimde çalışan programları ayırt etmemizi sağlar)

pgrep -lu username → kullanıcıya ait uygulama isimlerini ve PID lerini gösterir

programadı → girilen programı çalıştırır, ctrl + c ile durdurur

programadı & → girilen programı arka planda çalıştırır

jobs → arkaplanda çalışan programaları gösterir

fg → arkaplanda çalışan uygulamaya geçiş yapmayı sağlar

wc dosyaadı → dosya içindeki satır, kelime ve karakter bilgilerini sırasıyla gösterir

lspci → sistemin donanım bilgilerini gösterir

pwgen -sy sayı → girilen sayı kadar rastgele şifre oluşturur

xinput -l → gerçekten çıkarılabilir donanımları listeler

uptime → sisteme ait çalışma süresi bilgisi ve kullanıcı sayısını gösterir

nano metindosyaadı → girilen metin dosyasını terminalde açıp düzenlemeye yarar

nano .bash_history → kullanıcıya ait .bash_history metin dosyasını terminalde açar

cat .bash_history | grep ls → .bash_history metin dosyası içerisinde ls geçen kelimeleri gösterir

cat .bash_history | grep ls >> ls-ler → .bash_history metin dosyası içerisinde ls geçen kelimeleri ls-ler adlı dosyaya yazdırır

İZİN VERME

ls -al ile dosyaların indekslerini görüntülemiştik. Bu indekslerden biri de kullanıcı izinleridir.

-rw----- 1 mehmet mehmet 11052 Ağu 11 15:37 .bash_history

-rw-rw-r-- 1 mehmet mehmet 1024 Ağu 11 15:35 ..bash_history.swp

drwxr-xr-x 2 mehmet mehmet 4096 Ağu 9 20:28 Desktop

başında **d** olanlar dizinleri, - olanlar dosyaları ifade eder, **d** ve - den sonraki 9 lu sıralama ise ;

ilk üçlü : **user**(kullanıcı) izinleri, ikinci üçlü: **group** izinleri, üçüncü üçlü: **other**(diğer kişilere verilen) izinleri temsil eder

x: exploitation(çalıştırma izni) **w**: write(yazma izni) **r**: read(okuma izni)
chmod u+x dosyaadı → girilen dosyaya u (user-kullanıcı) için x (exploitation-çalıştırma) izni verir

```
mehmet@m:~$ ls -al mehmet
-rw-rw-r-- 1 mehmet mehmet 0 Ağu 11 15:52 mehmet
```

```
mehmet@m:~$ chmod u+x mehmet
```

```
mehmet@m:~$ ls -al mehmet
-rwxrw-r-- 1 mehmet mehmet 0 Ağu 11 15:52 mehmet
```

chmod u-x dosyaadı → girilen dosyaya u (user-kullanıcı) için x (exploitation-çalıştırma) iznini geri alır

```
mehmet@m:~$ chmod u-x mehmet
```

```
mehmet@m:~$ ls -al mehmet
-rw-rw-r-- 1 mehmet mehmet 0 Ağu 11 15:52 mehmet
```

chmod ugo+wr dosyaadı → u(user-kullanıcı), g(grup) ve o(other-diğer) için yazma ve okuma izni verir

```
mehmet@m:~$ ls -al mehmet
----- 1 mehmet mehmet 0 Ağu 11 15:52 mehmet
```

```
mehmet@m:~$ chmod ugo+xr mehmet
```

```
mehmet@m:~$ ls -al mehmet
-r-xr-xr-x 1 mehmet mehmet 0 Ağu 11 15:52 mehmet
```

Rakamsal Kodlarla izin verme;

r → 4 **w** → 2 **x** → 1

örneğin ; rw - r - - r - - => 644

rw x r - x - - - => 750

r - x r - - rw x => 547

rw x rw x rw x => 777 gibi...

chmod 777 dosyaadı → tüm kullanıcılar için tüm izinleri verir

```
mehmet@m:~$ chmod 777 mehmet
```

```
mehmet@m:~$ ls -al mehmet
-rwxrwxrwx 1 mehmet mehmet 0 Ağu 11 15:52 mehmet
```


chmod 000 dosyaadı : tüm izinleri alır

```
mehmet@m:~$ chmod 000 mehmet
```

```
mehmet@m:~$ ls -al mehmet
```

```
----- 1 mehmet mehmet 0 Ağu 11 15:52 mehmet
```

chmod -reference=ilkdosya ikincidosya → ilk dosyanın izinlerinin aynısını ikinci dosyaya da verir

```
mehmet@m:~$ ls -al dosya1 dosya2
```

```
-rw-rw-r-- 1 mehmet mehmet 0 Ağu 11 16:06 dosya1
```

```
----- 1 mehmet mehmet 0 Ağu 11 16:08 dosya2
```

```
mehmet@m:~$ chmod -reference=dosya1 dosya2
```

```
mehmet@m:~/linux$ ls -al dosya1 dosya2
```

```
-rw-rw-r-- 1 mehmet mehmet 0 Ağu 11 16:06 dosya1
```

```
----- 1 mehmet mehmet 0 Ağu 11 16:08 dosya2
```

chmod u-w * → dizin içindeki tüm dosyalardan **u** (user-kullanıcı) için yazma iznini alır

```
mehmet@m:~$ ls -al dosya1 dosya2
```

```
-rw-rw-r-- 1 mehmet mehmet 0 Ağu 11 16:15 dosya1
```

```
-rw-rw-r-- 1 mehmet mehmet 0 Ağu 11 16:15 dosya2
```

```
mehmet@m:~$ chmod u-w *
```

```
mehmet@m:~$ ls -al dosya1 dosya2
```

```
-r--rw-r-- 1 mehmet mehmet 0 Ağu 11 16:15 dosya1
```

```
-r--rw-r-- 1 mehmet mehmet 0 Ağu 11 16:15 dosya2
```

chmod ugo+x *.txt → sonu .txt ile biten her dosya için u(user-kullanıcı), g(gruop) ve o(other-diğer) için çalıştırma izni ekler

```
mehmet@m:~$ ls -al metin1.txt metin2.txt
```

```
-rw-rw-r-- 1 mehmet mehmet 0 Ağu 11 16:20 metin1.txt
```

```
-rw-rw-r-- 1 mehmet mehmet 0 Ağu 11 16:20 metin2.txt
```

```
mehmet@m:~$ chmod ugo+x *.txt
```

```
mehmet@m:~$ ls -al metin1.txt metin2.txt
```

```
-rwxrwxr-x 1 mehmet mehmet 0 Ağu 11 16:20 metin1.txt
```

```
-rwxrwxr-x 1 mehmet mehmet 0 Ağu 11 16:20 metin2.txt
```

-rwsr-xr-x 1 root root 54256 Mar 29 12:25 /usr/bin/passwd → **s**; suid yetkisi. Root yetkilerine sahip olarak çalıştırır. Passwd yazarak parolamızı değiştirirken root yetkisi istemediği gibi...

grup kısmında s olursa (sgid tetkisi) → o grup olarak çalıştırma yetkisi verilmiş olur

umask → yeni oluşturulacak dosyanın izinlerini ayarlamaya yarar

```
mehmet@ubuntu:~$ umask
```

```
0002
```

```
mehmet@ubuntu:~$ touch dosya
```

```
mehmet@ubuntu:~$ ls -l dosya
```

```
-rw-rw-r-- 1 mehmet mehmet 0 Ağu 17 15:03 dosya
```

hesaplaması → normal durumda verilebilecek (çalıştırılabilir olması olası) 666. 0666-0002= 0664
yani -rw - rw- r - - dir.

KULLANICI EKLEME – SİLME

sudo useradd newuser → girilen isimde yeni kullanıcı ekle

sudo passwd newuser → yeni kullanıcı için parola belirlenmesini sağlar

sudo userdel newuser → girilen kullanıcı ismindeki kullanıcıyı siler

sudo adduser newuser → yeni kullanıcı oluşturmayı sağlar ve yeni kullanıcı için parola girilmesi istenir

sudo deluser - - remove-all-files newuser → yeni kullanıcıyı silerken tüm dosyaları ile birlikte siler

su - newuser → kullanıcılar arası geçiş yapmayı sağlar

adduser username sudo → kullanıcıyı sudoers grubuna ekler(root yetkisi verir)

sudo su newuser → kullanıcı geçişi sağlar

id → sistemdeki kullanıcıları gösterir

cat /etc/group → sisteme ait grupları gösterir (yazıcı grupları,tarayıcı grupları, kullanıcı grupları vb.)

groups kullanıcıadı → kullanıcı adının mevcut olduğu grupları gösterir

```
mehmet@ubuntu:~$ groups mehmet
```

```
mehmet : mehmet adm cdrom sudo dip plugdev lpadmin sambashare
```

cat /etc/passwd → kullanıcı bilgilerini ayrıntılı olarak gösterir

usermod -aG sudo username → sudo grubuna girilen kullanıcıyı ekler (kullanıcı root yetkisine sahip değildir. Sadece sudo yetkisi verir)

PAKET YÖNETİMİ

apt- → paket yöneticisinin ismi

paketleri indirir, düzenler, bilgilerini tutar, günceller, yükseltir ve paketi siler

sudo apt-get update → paket listesi güncellenir

sudo apt-get upgrade → güncellenen paket listesindeki paketleri yükseltir

sudo apt-get dist-upgrade → sistemde yüklü mevcut linux dağıtımının yeni sürümü çıkmış ise sistem yeni sürüme güncellenir

sudo apt-get update && apt-get upgrade → önce paket listesi güncellenir sonra paketler yükseltilir

sudo apt-get install paketadi → istenilen paketi yüklememizi sağlar

sudo apt-get purge paketadi → paket ile gelen her türlü bağlantılı dosyayı siler

sudo apt-get remove paketadi → sadece paketi siler (paket ile ilgili bağlantılı dosyalar kalır)

sudo apt-get clean → cache'in önbelleğinde bulunanları siler

sudo apt-get autoclean → önbellekte bulunan gereksiz dosyaları siler

sudo apt-get autoremove → adı bilinmeyen, eskimiş paket vs. gibi kullanılmayan dosyaları otomatik siler

sudo apt-cache pkgnames → var olan paketlerin isimlerini listeler

sudo apt-cache pkgnames | wc → var olan paketlerin sayısını gösterir

sudo apt-cache show isim → girilen ismi içeren paketleri gösterir

sudo nano /etc/apt/source.list → kaynak listesini terminalde metin düzenleyici olarak açar

strace komutadi → komutun neler yaptığıyla ilgili bilgiler verir

ldd /bin/nano → nano komutunun bağlı olduğu kitaplıkları gösterir

dpkg komutuyla **.deb** uzantılı paketlerin durumunu ve bu paket kurulması kaldırılması gibi basit işlemleri de gerçekleştirebilirsiniz. Komut aracılığıyla sistem genelinde işlemleri yapabilmek için **sudo** komutu ile birlikte kullanabilirsiniz..

dpkg -l | more → mevcut depo paketlerini listeler

dpkg -l | grep paketadi → girilen paket adına ait sistemde yüklü paketleri listeler

dpkg -i paketadi → girilen paket adına ait paketleri yüklemeye başlar

dpkg -R /kurulacakpaketinyolu → girilen yolda bulunan paketi kurar

dpkg -r paketadi → adi girilen paketi siler

dpkg -P paketadı → adı girilen pakete ait tüm yapılandırma dosyalarını siler

dpkg -l paketadı → girilen paketle ilgili gerekli bilgileri listeler

dpkg -l *adi* → girilen ad geçen paketleri listeler

dpkg -L paketadı → aramak istediğimiz paket nerelere yüklendiğini gösterir

dpkg -p paketadı → paketle ilgili daha detaylı bilgi verir

dpkg -s paketadı | grep Status → paketin durumu (status) ile ilgili bilgi verir

mehmet@m:~\$ dpkg -s less | grep Status

Status: install ok installed

dpkg -c paketadı.deb → deb paketinin kim tarafından yazıldığını gösterir

mehmet@ubuntu:~\$ **dpkg -S \$(which ls)** → komuta ait paketi bulur

coreutils: /bin/ls

ARŞİVLEME – SIKIŞTIRMA

tar → arşivleme işlemi için kullanılır. Herhangi bir boyut değişikliği olmaz

zip yeniad.zip sıkıştırılacakdosyanınadı → sıkıştırıldıktan sonraki ismi .zip olarak girildikten sonra sıkıştırılacak dosyanın yolu belirtilir. Eğer aynı dizin içindeyse yol belirtmeye gerek yoktur.

zip yeniad.zip *.txt → sonu .txt ile biten dosyaları girilen yeniad.zip dosyasına sıkıştırır

zip -r yenedizinadı.zip dizinadı → belirtilen dizini belirtilen yenedizinadı.zip olarak sıkıştırır

unzip ziplenendosyaadı.zip zipiçindekidosya → unzip ten sonra bir zip dosyası adı ve içindeki bir dosya adı girilirse, sadece girilen dosya zipten çıkarılır

tar operasyon opsiyon

c → creat(oluşturma)

x → extract (ayırma)

t → rapor

v → verbose (ayrıntılı)

f → file (dosya)

z → zip(sıkıştırma)

tar xvf dosyaadı.tar → arşivlenen dosya çıkarılır

tar czvf dosyaadı.tar.gz içindeki ziplenecekdosyalar → ziplenecek dosyaları girilen dosyaadı.tar.gz olarak sıkıştırır

tar tzvf dosyaadı.tar.gz içindeki ziplenmişdosyalar → dosyaadı.tar.gz içerisindeki dosyaları gösterir

tar xzvf dosyaadı.tar.gz ziplenmişdosyalar → sıkıştırılmış dosyaları çıkartır

tar rvf dosyaadı.tar belirlibirdosya → belirtilen belirli bir dosyayı dosyaadı.tar içerisine arşivler

sıkıştırma**çıkartma****gzip dosyaadı
bzip2****gzip-d dosyaadı ya da gunzip dosyaadı
bunzip2 dosyaadı**

w parametresi → paket deposundan gelen rar dosyası gerçekmi değilmi kontrol etmemizi sağlar

tar tvfw dosyaadı.tar → girilen dosyanın tar doysası olup olmadığını kontrol etmemizi sağlar

tar cfvj dosyaadı.tar.bz2 dosyalar → girilen dosyaları bzip2 (.bz2) formatında sıkıştırmamızı sağlar

zip -er dosyaadı.zip ziylenecekdosyalar → girilen dosyaları sıkıştırır ve bu sıkıştırma dosyası için parola belirlememizi ister

mehmet@m:~\$ zip -er dosya.zip d1 d2

Enter password:

zip -P girileceparola dosyaadı.zip ziylenecekdosyalar → belirlenen dosyaları sıkıştırmamı sağlar. Burada girilecek parola açık olarak görünür.

mehmet@m:~\$ zip -P XDG123 dosya.zip d1 d2 d3 d4

unzip dosyaadı.zip → zip parolasını kaldırmamızı sağlar

ARAMA -TARAMA

locate → hızlı bir arama türüdür. Bunun nedeni daha önceden kullanılmış dosyaların yollarını sisteme kaydeden metin belgesinden almasıdır

locate dosyaadı → girilen dosyanın yolunu bulmamızı sağlar

mehmet@m:~\$ locate pwd
/bin/pwd

locate -i dosyaadı → arama yaparken küçük-büyük harf ayrımı yapmaz

mehmet@m:~\$ locate PWD

mehmet@m:~\$ locate -i PWd
/bin/pwd

sudo updatedb → yapılan son değişiklikleri kendi veritabanına ekler. Bu sayade locate ile arama yapmak kolaylaşır.

find

find yol ismi opsiyon/parametre çalıştırılması gereken komut

find /belirlenenyol -name “*l*” → girilen yol/dizin içinde bulunan, içinde l harfi geçen dosyaları arar

```
mehmet@m:~$ find /etc -name "[*]"
/etc/init.d/plymouth-log
/etc/init.d/resolvconf
/etc/ssh/moduli
/etc/lsb-rel
.....
```

find /belirlenenyol -type f -size -10K → 10 kb'dan küçük dosyaları için arama yapar

find /belirlenenyol -ctime -2 → son 2 gün içinde değişiklik yapılmış dosyalar için arama yapar

find /belirlenenyol -user kullanıcıadı → kullanıcıya ait dosyalar için arama yapar

find /home -user kullanıcıadı -exec /bin/rm {} ";" → kullanıcı sahibini siler

find /etc/apt -iname "source.list" → büyük-küçük harf ayrımı yapmadan source.list için /etc/apt içerisinde arama yapar

find /var -name log -type d → /var dizini içinde log kelimesi geçen dizinleri arar, **d** parametresi burada (directory) dizini temsil eder

find /bin -type f -executable → /bin dizini içinde çalıştırılabilir dosyaları bulur

find /bin -type f -executable | grep cat → /bin dizini içinde cat kelimesi geçen çalıştırabilir dosyaları bulur

find /bin -type f -executable | wc -l → /bin dizininin içinde çalışan kaç dosya olduğunun sayısını bulur

yukarıdaki wc -l parametresine ek olarak ;

wc -w → kelime bulma parametresi

wc -c → karakter bulma parametresi

find /home -type d -exec rmdir --ignore-tail-on-non-empty {} +; → bütün mevcut boş dizinleri bulur ve siler

esc + l → nano içerisinde bir metin yazarken; uzun kelimeleri kırarak alt satıra atmamızı sağlar

sort yazıadı → yazı içerisindeki satırları alfabetik sıraya göre sıralar

sort -u yazıadı → yazı içerisindeki satırları alfabetik olarak sıralarken tekrar eden satır başlarını tekrar sıralamaz

sort -n yazıadı → metin içindeki sayıları sıraya göre sıralar

sort -ru yazıadı → tekrar edenleri almadan, yazıyı ters çevirir

sort -k 3 yazıadı → her cümlemin 3. kelimesine göre sıralama yapar

sort -n d1 d2 d3 → **baskadosya** → karmaşık olan dosyaları başka bir dosyaya sıralar

```
mehmet@m:~$ cat d1
1 mehmet
5 burak
mehmet@m:~$ cat d2
3 kerem
4 nermin
```

```
mehmet@m:~$ cat d3
9 mert
6 aslı
```

```
mehmet@m:~$ cat baskadosya
1 mehmet
3 kerem
4 nermin
5 burak
6 aslı
9 mert
```

sudo tail -f /var/log/auth.log → auth.log ile ilgili giriş-çıkış gibi her türlü işlemi canlı olarak görmemizi sağlar

sudo tail /var/log/kern.log → f parametresiz kullanmazsak, son 10 işlem ile ilgili giriş-çıkış gibi her türlü işlemi canlı olarak görmemizi sağlar

sudo tail -n 3 /var/log/kern.log → son 3 işlemi güncel olarak görmemizi sağlar

sudo tail -v /var/log/kern.log → -v parametresi, görüntülenen dosyanın adını başta görmemizi sağlar

sudo tail -c 100 /var/log/kern.log → kern.log ile ilgili son 100 baytlık işlemleri güncel olarak gösterir

sudo tail +c 150 /var/log/kern.log → kern.log ile ilgili 150 bayt üzeri işlemleri güncel olarak gösterir

sudo tail -n 3 -f /var/log/kern.log → kern.log ile ilgili son 3 işlemi güncel olarak görmemizi sağlar

sudo tail /var/log/kern.log /var/log/syslog → her iki log dosyasını da güncel olarak görmemizi sağlar

sudo tail -f -s 5 /var/log/syslog → 5 sn içinde uykuda olan syslog işlemlerini gösterir

head → başlıkları görüntülemek için kullanılır

head -n 3 dosyaadı → dosya içerisindeki baştan 3 satırı görüntüler

head -c 5 dosyaadı → dosya içerisinde baştan 5 karakteri görüntüler

head -n 3 dosyaadı | sort -r >> dosya.txt → dosyaadı içindeki ilk 3 satırı ters çevirerek dosya.txt içine kaydeder

more +/"gnome"/var/log/syslog → syslog işlemlerinde gnome geçen işlemleri bulur

sudo gparted- → sistemde bulunan tüm disklerin özelliklerini gösterir

uname -r → sistemin kernel sürümünü gösterir

sudo nano /etc/fstab → dosya sistemlerimizle ilgili bilgileri terminal metin editörü ile görüntülememizi sağlar

sudo fdisk -l → sistemdeki diskler ile ilgili daha ayrıntılı bilgiler verir

xdg-open /açılmakistenendizininyolu → açılmak istenen dizinin terminalden açılmasını sağlar

sudo mount /dev/sbd1 /media/dizin/ → sisteme sonradan takılan medya aygıtını (sbd1) /media/dizin/ dizini içine mount(bağlamamızı) etmemizi sağlar

sudo umount /media/dizin/ → mount etme (bağlama) işlemini geri alır

sudo apt update && sudo apt upgrade

sudo apt install pyhthon-pip

sudo pip install cheat

pip install youtube-dl → youtube-dl 'ın son sürümünü yükler

youtube-dl youtubelinki → girilen youtube linkindeki videoyu indirir

TAKMA ADLAR

alias yenikomut='mevcutkomut'

alias 'ı iptal etmek için; **unalias yenikomut**

önce **nano /home/mehmet/.bashrc** ile metin çine girip alias ların olduğu satırlardan birinin altına **alias yenikomut='mevcutkomut'** yazıyoruz. Sonra çıkıp **source /home/mehmet/.bashrc** yazarak .bashrc dosyasını sistemde güncelliyoruz.

lspci → mevcut cihazları listeler

lspci -k → icihazlara ait bilgiler verir

lspci -v → tüm cihazlar hakkında daha ayrıntılı bilgi verir

du -h /home/mehmet → kb cinsinden /home/mehmet dosyasının içindekilerin boyutunu gösterir

whereis dosya → belirlenen dosyanın dizinini gösterir

cat > here <<son

>yazi....

>yazi...

çıkmaq için → **son**

GNU-LINUX DEBİAN TERMİNAL KOMUTLARI 2

konu dışı .. """"mutt ya da ELM terminal üzerinden e posta istemcileridir""""

"" notifi-send yazı → girilen yazıyı kenardan bildirim olarak gösterir""

""notify-send -icon=/home/mehmet/mehemt.jpg Hoşgeldiniz → kenarda bildirim verirken yanına foto da koyar""

alias 1='notify-send "Mehmet" --icon=/home/mehmet/asd.jpg "mekanına hos geldin"

date -d “yıl-gün-ay saat-dakika-saniye” “+%s” → girilen zamandan bu güne geçen unix time ‘1 gösterir

```
mehmet@m:~$ date -d "1995-09-29 12:30:02" "+%s"
812370602
```

date -d @unixtime → girilen unix time’a ait bilgileri gösterir

```
mehmet@m:~$ date -d @812370602
Cum Eyl 29 12:30:02 EET 1995
```

ZAMANLANMIŞ GÖREVLER

crontab → belirli bi zaman aralığında bi şeyin çalışmasını istiyorsak cron komutu kullanılır. Cron bir yapılandırma dosyasıdır. /etc dizininin altında bulunur.

sudo nano /etc/crontab → zamanlanmış görevlerin listesini terminalde metin düzenleyici olarak açar

```
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.
```

```
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
```

```
# m h dom mon dow user  command
17 * * * * root    cd / && run-parts --report /etc/cron.hourly
25 6 * * * root    test -x /usr/sbin/anacron || ( cd / && run-par$
47 6 * * 7 root    test -x /usr/sbin/anacron || ( cd / && run-par$
52 6 1 * * root    test -x /usr/sbin/anacron || ( cd / && run-par$
#
```

dakika – saat - ayın günleri – ay - haftanın günleri
52 6 1 * *

crontab -e → crontab’ı mevcut editörle açmayı sağlar

export EDITOR="nano" → mevcut editörü nano olarak ayarlar

sudo nano /etc/crontab → zamanlanmış görev eklemek için bu dosyanın içine girilir

m h dom mon dow user command

32 12 * * * mehmet ls -lah /home >>/home/mehmet/linux/listeleme.tx → hergün saat 12 32 de /home dizinini listeleyip linux dizininin içine listeleme.txt olarak kaydeder

sudo service crond stop → zamanlanmış görevleri durdurur

sudo service crond start → zamanlanmış görevleri başlatır

/etc/init.d/cron.d start → zamanlanmış görevleri başlatır

at saat:dakika → at saat:dakika girildikten sonra yapılacak komutların girilmesi istenir.
ctrl + d çıkılır.

at now → şimdi çalıştırılacak olan komutları çalıştırır

atrm sayı → zamanlanmış görevlerde bulunan işlem sırasında yer alan görevlerin sayısına göre iptal eder

at now + 2 minutes → girildikten sonra 2 dk sonrası için görev eklememizi sağlar

at 15:30 tomorrow → girilen komutu yarın 15:30 da çalıştırır

at 03:00am → girilen öğleden sonra 3 de çalışır

at 05:00 Aug 16,25 → girilen görev agustosun 16 sı ve 25 inde 05.00 da çalışır

grep [harf...] dosyaadı → belirtilen dosyanın içinde bulunan kelimelerde, girilen harfleri içeren kelimeleri listeler

```
mehmet@m:~$ grep [m] d1
kerem
mehmet
```

grep [harf1...]harf2 dosyaadı → belirtilen dosya içerisinde girilen harf1 leri içeren ve sonu harf2 ile biten kelimeleri listeler

grep b.n dosyaadı → girilen dosya adının içinde, b ve n geçen , ortasındaki harfin fark etmediği isimleri listeler

grep f...r dosyaadı → başında sonunda f ve r olan 5 harfli kelimeleri listeler

```
mehmet@m:~$ grep f...r d3
fakir fankr fundr
```

grep ^harf dosyaadı → girilen dosyaadındaki ^harf yazarak başlardaki harfler geçen kelimeleri bulabiliriz

grep harf\$ dosyaadı → girilen dosyaadındaki harfi yazarsak, son dizelerinde girilen harfler geçen kelimeleri listeler

grep x\\\$ dosyaadı → girilen dosyadaki x\$ geçen ifadeleri içeren kelimeleri bulur

[: alpha :] → içinde alfabeye ait rakamaları bulur

[: digit :] → tek basamaklı rakamları bulur

[: space :] → boşluk karakterlerini bulur

Sed - Stream Editor For Filtering And Transforming Text

sed → kelime değişimleri için kullanılan komuttur

s → switch (değiştir) parametresidir

sed s/değiştirilecek/yerinegetirilecek/ → sed s/ dedikten sonradeki değiştirilmek istenen kelime vb. /yerine getirilecek kelime vb. Yazılır ve değişim gerçekleşir

```
mehmet@m:~$ cat dosya
benim adım x
x benim adımdır
x is my name
```

```
mehmet@m:~$ sed s/x/mehmet/ dosya
benim adım mehmet
mehmet benim adımdır
mehmet is my name
```

sed s/değiştirilecek/yerinegetirilecek/ >>dosya.txt → yapılan işlemi dosya.txt metin dosyasına kaydeder

sed s/değiştirilecek/yerinegetirilecek/ -i dosya.txt → yapılan işlemi dosya.txt metin dosyasına kaydeder

sed s/değiştirilecek/yerinegetirilecek/g -i dosya.txt → g parametresi de kullanılırsa değiştirilecek harf vb. Bulunan tüm satırları değiştirir

```
mehmet@m:~/$ cat dosya5.txt
benim adım hmet
benim adım mehhmet
benim adım meehmet
benim adım meeeehmet
```

```
mehmet@m:~$ sed s/[a-z]*hmet/mehmet/g dosya5.txt
benim adım mehmet
benim adım mehmet
benim adım mehmet
benim adım mehmet
```

s/ den sonra [] köşeli parantez kullanmak için iki harf gerekir. [me] gibi...

```
mehmet@m:~$ cat dosya6.txt
hmet
aaahmet
aaaaahmet
aaaaaaaahmet
```

```
mehmet@m:~$ sed s/a*hmet/ahmet/g dosya6.txt
ahmet
ahmet
ahmet
ahmet
```

satır değiştirme

sed /satırsayısı s/değiştirilecek/yerinegetirilecek/ → değiştirilmek istenen satır girilir ve işlem gerçekleştirilir

nano -c dosyaadı → girilen dosyaadının içindeki metinlerin satır sayısını metin editörü içinde öğrenmemizi sağlar

nano içerisinde alt + g ile hangi satıra gitmek istenirse gidilebilir

sed -e s/x/y/g -e s/z/t/g dosyaadı → girilen dosya içindeki metinde; x ler y olarak, z ler de t olarak değiştirilir.

```
mehmet@m:~$ cat dosya
benim adım x
x benim adım
x is my name
```

```
mehmet@m:~$ sed -e s/x/555/g -e s/benim/senin/g
senin adın 555
555 senin adın
555 is my name
```

sed s/^/"kelime "/g dosyaadı → belirlenen dosya içerisindeki her satır başına belirlenen kelime eklenir

```
mehmet@m:~$ cat liste
mehmet kılıç
serra salim
okan aydar
```

```
mehmet@m:~$ sed s/^/"sayın "/g liste
sayın mehmet kılıç
sayın serra salim
sayın okan aydar
```

mehmet@m:~\$ sed s/./"sayın "&/g liste → **#başında herhangi bir karakter varsa ya da yoksa bizim verdiğimiz metni satır başına ekle**

sayın mehmet kılıç
sayın serra salim
sayın okan aydar

awk → bir metin içerisinde satırlara değişken atayabilmemizi sağlar

mehmet@m:~/\$ awk '{ print \$1 }' liste → **sadece birinci satırı gösterir**

mehmet
serra
okan

mehmet@m:~\$ awk '{ print \$1" test txt "\$2}' liste → **1. ve 3. satır arasına girilen metin yazılır**

mehmet test txt kılıç
serra test txt salim
okan test txt aydar

mehmet@m:~\$ cat liste

sayın mehmet kılıç
sayın serra salim
sayın okan aydar

mehmet@m:~\$ awk '/sayın/{print \$2"-"\$3}' liste → **sayın kelimesi silinir; 2. ve 3. satırlar değişkene atanır. 2. ve 3. satırlar arasına – koyulur ve ekrana yazdırılır.**

mehmet-kılıç
serra-salim
okan-aydar

mehmet@m:~\$ pwd | awk '{print "merhaba mehmet, şu anda "\$1" dizininde sin }'
merhaba mehmet, şu anda /home/mehet/linux dizininde sin

üstte bulunan komutu alias ile “neredeyim” komutuna dönüştürelim.

Öncelikle bir metin dosyası içerisine pwd | awk '{print "merhaba mehmet, şu anda "\$1" dizininde sin }' yazalım. Sonra nano /home/username/.bashrc içerisinde girelim ve alias'ların bulunduğu yere alias neredeyim=' bash /home/mehmet/dosyaadı '

mehmet@m:~/linux\$ neredeyim
merhaba mehmet, şu anda /home/mehmet/linux dizininde sin

ya da Sonra nano /home/username/.bashrc içerisine

alias neredeyim=' pwd | awk '\{print "şu anda "\$1" dizininde sin seni lanet olası"}'\ yazabiliriz.

mehmet@m:~\$ uyduruk="deneme metni, burası bolu"
mehmet@m:~\$ echo \$uyduruk
deneme metni, burası bolu

nano test.sh dosyası oluşturup içine;

```
echo bir sayı giriniz
read sayi
echo girilen sayi $sayi
```

```
mehmet@m:~$ bash test.sh
bir sayı giriniz
456
girilen sayi 456
```

```
echo bulunulan dizin
read sayi
echo dizinde bulunanlar $sayi
```

```
mehmet@m:~/linux$ bash test.sh
bulunulan dizin
*
dizinde bulunanlar crontab dosya dosya5.txt.save liste test.sh
```

```
mehmet@m:~/linux$ let "carpim=2*7"
mehmet@m:~/linux$ echo $carpim
14
```

nano ile carpim.sh dosyası içerisine aşağıdaki kodları yazıyoruz;

```
echo  çarpımak istediğiniz iki sayıyı sırası ile giriniz
read sayi1
read sayi2
let  carpim=sayi1*sayi2
echo  $sayi1 ile $sayi2 nin çarpımı= $carpim
```

```
mehmet@m:~/linux$ bash carpim.sh
çarpımak istediğiniz iki sayıyı sırası ile giriniz
5
6
5 ile 6 nin çarpımı= 30
```

```
#!/bin/bash
```

```
echo  2 sayı ile çarpım işlemi ekranına hoşgeldiniz
echo  ilk sayiyi giriniz
read  num1
echo  ikinci sayiyi giriniz
read  num2
echo  "$num1 * $num2" | bc -l
```

BASH SCRIPT'DE İF YAPISI

```
if gnu/linux komutu
    then
        komut1
        komut2
    else
        komut3
        komut4
        komut5
fi
```

terminal her açıldığında bulunduğu dizini listeyen program;

bir metin dosyası içine aşağıdaki kodları yazarak metin dosyasını /bin/ dizini içine atıyoruz.

```
#!/bin/bash
```

```
ls -l
```

daha sonra nano /home/mehmet/.bashrc metin içine aşağıdaki kodları ekliyoruz. Bu sayede terminal her açıldığında bulunduğu dizini listelemiş oluyor.

```
if [ -x /bin/ter ]; then
    ter
fi
```

```
-gt → büyüktür
-lt → küçüktür
-ge → büyük eşit
-le → küçük eşit
-eq → eşittir
-ne → eşit değil
```

mantıksal karşılaştırmalarda ;

```
-a → ve
-o → veya
! → değil
```

sayı tek basamaklı mı, çift basamaklı mı ?

```
#!/bin/bash
```

```
echo o ile 20 arasında bir sayı giriniz
read sayi
if [$sayi -ge 10 ];then
    echo $sayi çift basamaklıdır
else
    echo $sayi tek basamaklıdır
fi
```

case yapısı

özel karakterlerin içinde de kullanılabilir

```
#!/bin/bash
```

```
echo "1. seçenek=ekranı temizle"
```

```
echo "2. seçenek= bulunulan dizini listele"
```

```
echo -n "seçenek giriniz" → -n ; numeric (sayısal ) değerler girilmesi için kullanılır
```

```
read secenek
```

```
case $secenek in
```

```
1) clear
```

```
;;
```

```
2) ls .
```

```
;;
```

```
*)
```

```
echo "EROR!"
```

```
esac
```

while

birden yüze kadar sayan while döngüsü;

```
#!/bin/bash
```

```
i="1"
```

```
while [ $i -le 100 ]
```

```
do
```

```
echo $i
```

```
i=$((i+1))
```

```
done
```


14/08/2016 GNU-LINUX DEBİAN TERMİNAL KOMUTLARI 3 FOR YAPISI

```
#!/bin/bash
for degisken in d1 d2 d3
do
    komutlar
done
```

```
#!/bin/bash

for bilgisayar in cpu ram gpu
do
    echo $bilgisayar
done
```

```
mehmet@mehmet-Casper-Nirvana-Notebook:~$ bash ornek.sh
cpu
ram
gpu
```

```
#!/bin/bash
if [ $# = 1 ]
then
    ps aux | grep -i $1
else
    echo "bir program girilmedi"
fi
```

/usr/local/bin/ → kullanıcıların kendisi yazdığı bash dosyalarını bu dizine atması daha uygundur
sudo cp pid /usr/local/bin/

```
mehmet@m:~$ pid firefox
mehmet  5592  0.0  0.0 21024 3272 pts/11  S+  12:18  0:00 /bin/bash /usr/local/bin/pid firefox
mehmet  5594  0.0  0.0 22700  928 pts/11  S+  12:18  0:00 grep -i firefox
```

internet → inter networks ; ağlar arası ağlar
internet otonom çalışır. Gayri merkezidir.

Ip → internet protokol ; ağ işlemlerinin gerçekleştirildiği katman anlamına gelir

ağ cihazlara ip paketlerini istenilen adreslere yönlendirilmesini sağlar
ip ile uygulamalar arasında protokolü sağlayan ;

1-tcp → transmission control protokol; tcpip → ağ üzerinden veri aktarımını sağlar, paket iletimlerinde hata kontrolü yapar, hataları düzeltir

2-udp → user datagram protokol, paketlerin iletiminde hata kontrolü gerektirmeyen işlemlerde kullanılır; genelde akış verileri, ses, görüntü, video işlemlerinde kullanılır

telnet → veriler şifrelenmez ; Internet ağı üzerindeki çok kullanıcı bir makineye uzaktaki başka bir makineden bağlanmak için geliştirilen bir TCP/IP protokolü ve bu işi yapan programlara verilen genel isimdir.

ssh → secure shell ;güvenli veri iletimi için kriptografik ağ protokolüdür.

Http 80 ; https 443 portunu kullanılır

http → "Hyper Text Transfer Protocol" (Hiper Metin Transferi Protokolü) kelimelerinin baş harflerinden oluşan bir kısaltmadır.

Https → HyperText Transfer Protocol Secure kelimelerinin baş harflerinden oluşan bir kısaltmadır. Hypertext terimi, bir web sitesinin kod veya eklenti gerektirmeyen, metin, tablo veya resim gibi içeriklerini açıklar.

/etc/services --> Bir gbu/linux sistemde portlara ait bilgiler /etc/services dosyasının içerisinde bulunur.

/etc/network → ağ ayarlarının bulunduğu dizin

systemd; init'in yeni halidir

networking → sistemde ağ var mı yokmu onu bulur

hostname → sistemimizin adını belirler

gateway → Ağ geçidi farklı ağ iletişim kurallarını kullanan iki bilgisayar ağı arasında veri çerçevelerinin iletimini sağlayan ağ donanımdır. Bir başka deyişle aynı dili konuşamayan iki ağ arasında tercüman vazifesi görür.

sudo ifconfig → ethernet ip adres bilgilerini öğrenmemizi sağlar

iss → isp -internet service provider; internet servis sağlayıcı

dns → domain name surver ;/etc/resolv.conf/ dns ayarları

dns değiştirmek için **nano /etc/resolv.conf** dosyasına girilir.

bcast → Gönderilen paketin ağ üzerinde bulunan herkese ulaşmasını sağlar

sudo ifconfig içerisindeki internet bağlantısı ayarlarından iptal etmek istenilen kablolu ya da kablosuz ağın kodu yazılır;

sudo ifconfig wlp2s0f0 down → interneti kapatır

sudo ifconfig wlp2s0f0 up → interneti açar11

ifconfig eth0 makinanınipsi networkipsi up → mevcut ip adresimizi sabitleyebilmek için kullanılır... örneğin ; **ifconfig eth0 10.47.150.9 255.255.0.0 up**

route → mevcut yönlendirme tablosu

sudo route add default gw ipadres → gateway değiştirir

15.08.2016 Monday Linux Yaz Kampı 2016 by MEHMET KILIÇ
GNU/LINUX DEBİAN TERMİNAL KOMUTLARI 4

DHCP(Dynamic Host Configuration Protocol) Internet Protokolüne (IP) bağlı çalışan ağlardaki hostlar (yani ağa bağlanan cihazlar) için bir ağ yapılandırma protokolüdür. İster yerel ağdaki bir bilgisayar, ister bir yazıcı ya da başka bir cihaz olsun, ağa bağlanan her cihazın bir [IP adresi](#) olmak zorundadır. IP ağlarına bağlanan cihazların, diğer hostlarla (yani ağa bağlı diğer cihazlarla) iletişim kurabilmeleri için yapılandırılmaları gerekir. Bu yapılandırmada ihtiyaç duyulan temel bilgi IP adresidir. DHCP, ağa bağlanmaya çalışan cihazlara otomatik olarak bir IP adres tahsisi yapar (dinamik IP adresi). Dhcp sunucusunu hayata geçiren servisin adı isc-dhcp-server'dır.

Public IP Address Classes range

Class	1st Octet DEC range	1st Octet BIN	Start address	Finish address	1st Octet High order Bits	Network/ Host	Default Subnet Mask
A	1-126	00000001-01111110	0.0.0.0	126.255.255.255	0	N.H.H.H	255.0.0.0
B	128-191	10000000-10111111	128.0.0.0	191.255.255.255	10	N.N.H.H	255.255.0.0
C	192-223	11000000-11011111	192.0.0.0	223.255.255.255	110	N.N.N.H	255.255.255.0
D	224-239	11100000-11101111	224.0.0.0	239.255.255.255	1110		
E	240-255	11110000-11111111	240.0.0.0	254.255.255.255	11110		

Note: Class A address **127.0.0.0 - 127.255.255.255** cannot be used and is for **LOOPBACK** and diagnostic

Private IP Address Classes range

Class	1st Octet DEC range	1st Octet BIN	Start address	Finish address	1st Octet High order Bits	Network/ Host	Default Subnet Mask
A	10	00001010	10.0.0.0	10.255.255.255	0	N.H.H.H	255.0.0.0
B	172	10101100	172.16.0.0	172.31.255.255	10	N.N.H.H	255.255.0.0
C	192	11000000	192.168.0.0	192.168.255.255	110	N.N.N.H	255.255.255.0

www.ic.ims.hr

public ip →

1-126 → a class ip bloğu

128-191 → b class ip bloğu

192-223 → c class ip bloğu

224-239 → d class ip bloğu

240-255 → e class ip bloğu

not : d ve e class ip ler genellikle bilimsel araştırmalar ve deneyler için kullanılır

private ip →

a block: 10.0.0.0.

b block :172.168.0.0.

c block: 192.168.0.0

bu iplerin internette bir karşılığı yoktur. Bulunduğu network için geçerlidir.

127.0.0.1/8 → internette kullanılan böyle bir ip yoktur. Bu bilgisayarın local ip sidir.

Lan → local area networking

van → lan'lar bir araya gelerek van ağını oluşturur

Verilen ip ile netmask ipsi end (çarpım) işleminden geçirildiğinde network içinde dağıtılabilecek ip numaraları bulunabilir.. örneğin;

```
Address: 192.168.56.0      11000000.10101000.00111000. 00000000
Netmask: 255.255.255.0 = 24 11111111.11111111.11111111. 00000000
```

24 ün anlamı; netmask'teki ilk, ikinci ve üçüncü bitlerin 8 bit olması; $3*8=24$. bu buradan elde edilir.

Yani 192.168.56.0 dan 192.168.56.255 e kadar network içinde dağıtılabilecek ip lerdir. Network'un ilk ip'si network'un kendisini ifade eder. Bir istemciye yada bir kişiye bu ip tahsis edilemez. Örneğin 192.168.56.0 yukarıdaki network'un kendisini ifade eder. 192.168.56.255 yukarıdaki örneğin (networkun son ip si olduğu için) network'un broadcast ip sidir. Bilgisayarlar birbirleriyle haberleşmesi için broadcast/yayın yapmak zorundalardır. Bu ip de bir istemciye yada bir kişiye bu ip tahsis edilemez.

Yani yukarıdaki örnekte kullanılabilecek ip'ler 192.168.56.1 dan 192.168.56.254 kadar olanlardır.

Gateway → (gateway bir router yada bilgisayar olabilir.)ağ geçidi, o networkle aynı networkde bulunmalıdır. Ağa bağlanmamızı sağlar. Gateway'lere genellikle (kural olmamakla birlikte) o network un verilebilir ilk ip si hangisi ise o verilir. Bazen ise verilebilen son ip gateway cihazına verilir. Yukarıdaki örnekte gateway 192.168.56.254 ip adresine verilmiştir.

**linux tabanlı işletim sistemleri router olarak tanımlanabilir.

ipcalc ipnumarası → girilen ip numarasına ait netmask, network, broadcast numaralarını gösterir

Soru: 139.179.179.0 ip numarası verilmiştir. Bu ip yi ikiye bölüm iki ayrı network oluşturulması isteniyor...

139.179.179.0 'dan 139.179.179.127 ip numarasına kadar kullanılabilen ip leri hesaplamamızı sağlar:

```
mehmet@casper:~$ ipcalc 139.179.179.0/25
Address: 139.179.179.0      10001011.10110011.10110011.0 00000000
Netmask: 255.255.255.128 = 25 11111111.11111111.11111111.1 00000000
Network: 139.179.179.0/25   10001011.10110011.10110011.0 00000000
Broadcast: 139.179.179.127  10001011.10110011.10110011.0 11111111
Hosts/Net: 126              Class B
```

139.179.179.128/25 →

139.179.179.128 'dan 139.179.179.155 ip numarasına kadar kullanılabilen ip leri hesaplamamızı sağlar:

```
mehmet@casper:~$ ipcalc 139.179.179.128/25
Address: 139.179.179.128   10001011.10110011.10110011.1 00000000
Netmask: 255.255.255.128 = 25 11111111.11111111.11111111.1 00000000
Network: 139.179.179.128/25 10001011.10110011.10110011.1 00000000
Broadcast: 139.179.179.255 10001011.10110011.10110011.1 11111111
Hosts/Net: 126              Class B
```

route -nv → ağ geçidine ait ip numaralarını öğrenmemizi sağlar

```
mehmet@casper:~$ route -nv
```

Kernel IP routing table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
0.0.0.0	10.47.1.1	0.0.0.0	UG	600	0	0	wlp2s0f0
10.11.0.20	10.47.1.1	255.255.255.255	UGH	600	0	0	wlp2s0f0
10.47.0.0	0.0.0.0	255.255.0.0	U	600	0	0	wlp2s0f0
169.254.0.0	0.0.0.0	255.255.0.0	U	1000	0	0	wlp2s0f0

Not: gateway’da 0.0.0.0 ip adresi görünmesinin anlamı; aynı network’te bulunan birisinin kullanılan makineye ulaşabileceği anlamına gelir.

ping ipadres → girilen ip adresine ping atar

netstat → bilgisayarda açık olan portları gösterir

```
mehmet@ubuntu:~$ sudo netstat -ntlp
```

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State	PID/Program name
tcp	0	0	127.0.1.1:53	0.0.0.0:*	LISTEN	1537/dnsmasq
tcp	0	0	0.0.0.0:25	0.0.0.0:*	LISTEN	1062/master
tcp6	0	0	:::25	:::*	LISTEN	1062/master

Ağa Bağlama

→ dhcp bir ip atar... örneğin 192.168.56.0/24

→ isc-dhcp-server kurulur

→ gateway (ağ geçidi)

→ range (aralık) ; ip aralık dağıtımı ... örneğin 192.168.56.101 ile 192.168.56.255 e kadar kullanım aralığı belirlensin

cat /etc/services | grep ssh → ssh’a ait port numarasını öğrenmemizi sağlar

```
mehmet@ubuntu:~$ cat /etc/services | grep ssh
```

```
ssh          22/tcp          # SSH Remote Login Protocol
ssh          22/udp
```

Sanal Makina ile Net’e Bağlanma

1- Use Bridged Networking : ilk network tipimiz olan use bridged networking seçeneği fiziksel makinemizde bulunan interface kartımızla bridge (köprü) vazifesi görerek bizi sanal makinelerimizle yahut internete çıkmamıza olanak sağlar. Bridgeleme yönteminde sanal makinalarımız (sanal interfacemiz) bir gerçek makine gibi davranarak bizi sanal makinalarımızla ve dış dünyayla iletişime geçirir. Sanal makinelerimiz üzerinde oluşturulan sanal NIC lerin kendine özgü MAC adresleri ve İP adresleri mevcuttur. MAC adreslerinin dünya üzerinde aynısı bulunamaz.

2- Use Network Address Translation (NAT) : NAT kavramını açıklayarak başlayalım. Örneğin işyerimizde 1 adet modem olduğunu ve 10 adette client (istemci) bilgisayar olduğunu düşünelim. Bu bilgisayarlarımız bir switch bağlı switch te direk modeme tüm clientlerimiz bu modem üzerinden internete çıkarlar ve tek bir ip adresi kullanırlar hangi makine çıkarsa çıksın global ip adresleri aynıdır. Local ip adresleri modemden veya içeride bir DHCP sunucu üzerinde

tanımladığımız 192.168.2.2 şeklinde gibi işte 10 clientin aynı anda internete çıkmaları ve bir adet global ip kullanmaları bir NAT işlemidir. Oluşturduğumuz sanal makinada aynısını yapar. Direk NAT yaparak gerçek makineye bağlanır. Sebebi ise sanal makinenin ip adresi farklıdır. Örneğin 192.168.78.18 gibi farklı bir IP bloğu üzerindedir. Özetleyecek olursak sanal makinamız NAT yaparak dış dünyaya açılabilir.

3- Use Host-Only Networking : Üçüncü seçeneğimiz olan use host-only networking seçeneği özel bir ağ gibi davranır. Aynı switch üzerinde olan gerçek makinelerin birbiriyle konuşması gibi düşünebiliriz. İnternete çıkamaz sanal makineler birbiriyle konuşabilir. Ağ içerisinde bulunan fiziksel makine ilede konuşabilir. Fakat dış dünyayla iletişime geçemez.

4- Do Not Use a Network Connection : Son seçeneğimiz olan do not use a network connection kavramı sanal makine oluştururken Network Type olarak en son seçebileceğimiz bir seçenektir. Eğer sanal makinamızın özel veya başka bir ağa dahil olmasını istemiyorsak. Do not use a network connection seçeneğini seçebiliriz. Sanal makinamız sadece fişe takılan ve hiçbir makine ile iletişimi olmayan bir fiziksel makine olarak ta düşünebiliriz.

dhclient -v ağarayüzü → bir DHCP sunucusundan manual olarak ip almamızı sağlar

mehmet@ubuntu : ifconfig → burada çıkan bilgilerde sol köşede bulunan ethernet ya da wifi ağ arayüzü adı yukarıdaki portadı kısmına yazılır...örneğin
mehmet@ubuntu:~\$ dhclient -v wlp20f0

STATIC IP

ifconfig ağarayüzüadı ipadress netmaskipadress up → manual olarak anlık ip adresi sabitlemeye yarar. Sistem boot edildiğinde bu ip sabit kalmaz.

sudo nano /etc/network/interfaces → ağ arayüzlerinin yönetildiği dosyayı terminalde metin editörü olarak açmamızı sağlar

/etc/network/interfaces içerisinde **static ip belirlemek için** aşağıdaki gibi static yazılarak adresler girilir.

```
#the primary network interfaces
auto wlp20f0 → ağ arayüzü adı
iface wlp20f0 inet static
    adress      192.168.57.254
    netmask     255.255.255.0
    gateway     192.168.57.1
```

Not: Bu işlem yapılırsa; sistem boot edilse dahi aynı ip adresini verir. Yani ip her durumda static kalır.

service networking restart → ip yapılandırılması değiştirildikten sonra bu işlem yapılır ve ip resetlenir

ifconfig ayarüzadı down & ifconfig arayüzadı up → wifi'ı kapayıp açmamıza yarar

16.08.2016 Salı by MEHMET KILIÇ

GNU-LINUX DEBİAN TERMİNAL KOMUTLARI 5

SSH

(Secure Shell) güvenli veri iletimi için kriptografik ağ protokolüdür. Ssh ile ağa bağlı olan iki bilgisayar arasında veri aktarımı güvenlik kanalı üzerinden güvensiz bir ağda yapılır.

Ssh ile sunucuya bağlanılır. Open ssh; Ssh ile internete bağlanmak için oluşturulan servistir.

Ssh ile sunucuya bağlanmak için;

- Sunucuya Network üzerinden ulaşabilir durumda olmak
- Sunucu üzerinde kullanıcı adı-parola bilgilerinin bulunması

/home/username/.ssh dizini içerisinde private ve public key'ler bulunur. Bulunmadığı takdirde ;
cd /home/username ve **mkdir .ssh** dizini oluşturulur. Daha sonra **ssh-keygen** komutu ile public ve private key'ler oluşturulur.

sudo cat /root/.ssh/id_rsa.pub → public key'imizi görüntülememizi sağlar

```
mehmet@ubuntu:~/ssh$ sudo ssh-keygen
```

```
sudo: unable to resolve host ubuntu
```

```
Generating public/private rsa key pair.
```

```
Enter file in which to save the key (/root/.ssh/id_rsa):
```

```
Created directory '/root/.ssh'.
```

```
Enter passphrase (empty for no passphrase):
```

```
Enter same passphrase again:
```

```
Your identification has been saved in /root/.ssh/id_rsa.
```

```
Your public key has been saved in /root/.ssh/id_rsa.pub.
```

```
The key fingerprint is:
```

```
SHA256:q83+Ho5utltC060asDKwdRf0TVsZqaW157Ap9WJit4E root@ubuntu
```

```
The key's randomart image is:
```

```
+---[RSA 2048]----+
```

```
| . . o+ |
| . . o o= |
| . . o= . |
| o .o + .|
| .. o S . o B |
| + . = o .E B o|
| . o . + + . = + |
| o +oB . . |
| . =X=+ |
+----[SHA256]-----+
```

ssh-keygen -t rsa -b 4096 -C "username@host" → -t parametresi : type -b parametresi: bit

-C parametresi: yorum satırı...

sistem boot edildiğinde aynı public key'i alabilmemiz için kullanıcı adı ve host kısmına daha önce kullandığımız kullanıcı adı-host girilir. Bu sayede sistem yenilense dahi public key'imiz aynı kalır.

ssh-keygen -y → passphrase girerek yeniden public key oluşturmamızı sağlar

ssh-keygen -p → var olan passphrase'i güncellememizi sağlar

passphrase → private key'e ulaşmamızı sağlayan bir ön güvenlik (parola) önlemidir.

Not :Public key kaybolduğu zaman private key ile public key oluşturulabilir.
Ama private key kaybolduğu zaman public key ile private key oluşturulamaz.

Not :Private key'in izinleri değiştirilirse key'imiz güvensiz hale gelir!

Örnek : Bir bash script yazarak, otomatik olarak public ve private key oluşturmak;

```
mehmet@ubuntu:~$ nano sshh
```

```
#!/bin/bash
```

```
mkdir ~/sshh
```

```
ssh-keygen -t rsa -b 4096 -N "123456789" -f "~/sshh/id_rsa"
```

```
mehmet@ubuntu:~$ bash ssh
```

Generating public/private rsa key pair.

Your identification has been saved in /home/newuser/sshh/id_rsa.

Your public key has been saved in /home/newuser/sshh/id_rsa.pub.

...

sudo ssh-keygen -l → public key'e ait fingerprint'imizi listeler

```
mehmet@ubuntu:~/.ssh$ sudo ssh-keygen -l
```

```
sudo: unable to resolve host ubuntu
```

```
[sudo] password for mehmet:
```

```
Enter file in which the key is (/root/.ssh/id_rsa):
```

```
2048 SHA256:q83+Ho5utltC060asDKwdRf0TVsZqaW157Ap9WJit4E root@ubuntu (RSA)
```

Bağlanılacak iplere ait bilgiler **.ssh/config** dosyası içinde bulunur

ssh config dosyası

HOST dev

HostName ubuntu.com → domain

Port 22 → cat /etc/services | grep ssh ile bulunan port numarasını yazdık

User mehmet → kullanıcı adı

yukarıdaki bash script yazıldıktan sonra **scp bashdosyası dev:/path** işlemi yapılır.

ssh domainadı -p portnumarası → ssh ile ağa bağlanmak için el ile port numarası girmemizi sağlar

ping ipadresi → ssh ile bir yere bağlanmadan önce; aynı ağ üzerinde bulunan ip ye ping atılır. Ping atma işleminde 64 bytes 'lık veri paketleri ICMP (internet control message protocol) ile gönderilir.

ssh kullanıcıadı@bağlanılacakipadresi ve **ssh -l kullanıcıadı ipadres**i → ssh ile girilen ip adresine ait kullanıcıya bağlanmamızı sağlar

chmod 700 ~/.ssh/ → Bu dizinin içerisinde public ve private key olduğu için bu dizinin izinleri her zaman böyle olmalıdır. drwx - - - - - şeklindedir.

chmod 644 ~/.ssh/ip_rsa → Bu dosya private key olduğu için bu dizinin izinleri her zaman böyle olmalıdır. -rw-r - - r - - şeklindedir.

diff dosya1 dosya2 → girilen 2 dosyayı karşılaştırır. Benzerlik olan satır vb. Yerleri gösterir. Aynı olan kısımları ise göstermez.

scp → güvenli kopyalama komutudur

lsb_release -a → linux dağıtım sürümünü gösterir

mehmet@ubuntu:~\$ lsb_release -a

No LSB modules are available.

Distributor ID: Ubuntu

Description: Ubuntu 16.04 LTS

Release: 16.04

Codename: xenial

ssh domainaadı -i .ssh/id_rsa → girilen domaine bağlanmak için kullanılır. -i parametresi; birden fazla private key varsa hangisinin kullanılacağını belirtir.

echo \$? → bir önceki komutun çalışıp çalışmadığını gösterir. Sonuç 0 ise çalışmış; 0 değil ise çalışmamıştır.

nano /etc/ssh/sshd_config → ssh servisinin dosyasıdır. Burda yapılan bir işlem sonucu dosya yeniden başlatılmalıdır. Ssh ile ilgili yapılandırma ayarları buradadır

scp /home/mehmet/Videos/Atiye\ -\ Uyan\ Da\ Gel-NN9Sg77kpqY.mkv

[hakan@10.47.160.103:/home/hakan/Desktop](#) → scp(secure copy-paste) işlemi; aynı network içinde bulunan iki kişinin birbiriyle güvenli şekilde veri kopyalamasını sağlar. Komut yazılıp onaylandığında ise gönderilecek kişinin parolası girmesi gerekmektedir.

scp -r /home/mehmet/mehmet/ [sysadmin@10.47.185.202:](#) → bir yol belirlenmezse belirlenen izin default olarak kullanıcının ev dizinine kopyalar

/var/log\$ cat lastlog → uzaktan sisteminize erişen son kişinin ipsini tutar

w → sistemde çalışan kullanıcıları ve bilgilerini gösterir

ssh kullanıcıadı@ipnumarası → ile bağlanıldığı zaman; w komutu ile hangi ip nin bağlandığı görülebilir. **ps aux** kısmından bu bağlantıyı yapan sistemin ipsinin nin pid si bulunup **kill -9 pidno** yapılırsa bağlantı koparılmış olur.

killall sshd → ssh ile bağlanan tüm kullanıcıları sistemden atar

service ssh stop → ssh servisini durdurur

service ssh start → ssh servisini başlatır

simetrik şifreleme → ssh bağlantısı sırasında kullanılır(akan şifre); bağlantı açıldıktan sonra

asimetrik şifreleme → başta güvenlik(paralo doğrulama vb.) işler için ssh ın baş işlemi olarak kullanılır(blok şifre); key değişimi yapıldığı zaman-bağlantı açılmadan önce

traceroute ipadresı → girilen ip adresine hangi networkları kullanarak gidileceğini gösterir

VPN

Virtual Private Network (Sanal Özel Ağ) teriminin kısaltılmışı olan VPN, basit bir anlatımla internet üzerinde kurulu bilgisayarlar grubundan oluşur. Bu özel bilgisayar ağı, örneğin yerel bir ağa fiziksel erişimi bulunmayan, dışarıdaki bir kişi tarafından ağ kaynaklarına erişmekte kullanılabilir. VPN'ler, güvenilmeyen ağlara bağlanırken **bağlantıyı şifrelemek** ve güvenli hale getirmek için de kullanılır. Bunun yanında VPN'ler, farklı konumlarda bulunabildiğinden, sizi ülkelerin kısıtlamalarından (**yasaklı web siteleri** ve IP'ler gibi) da kurtarırlar.

VPN'lerle bağlantınız şifreli olduğundan, gönderdiğiniz ve aldığınız tüm veriler şifrelendiğinden, meraklı gözler ne yaptığınızı göremezler.

FTP

File Transfer Protocol (Dosya Transfer Protokolü) anlamına gelir, iki bilgisayar arasında dosya transferini sağlayan, internet ile beraber ilk geliştirilen protokoldür. Bir bilgisayardan diğer bilgisayara eş zamanlı dosya almak veya dosya çekmek için kullanılır. Güvenli bir transfer işlemi gerçekleşmez.

Apache2 ve **nginx** → Linux'ta en çok kullanılan web serverlarıdır

letsencrypt → ücretsiz ssl sertifikasıdır (ssl sertifikası ssh servisi ile çalışır)

/etc/ssl/certs → ssl sertifikaları bu dizinin altında bulunur, .pem uzantılı dosyalar ssl sertifika uzantısıdır. ssl sertifikaları sistemde gömülü ya da güncellemeler sırasında bilgisayarımıza otomatik olarak gelir. /etc/ssl/certs altında bulunan bir sertifikanın fingerprint'i; siteye girdiğimiz zaman sitede bulunan sertifikanın fingerprint'i ile eşleşir ve güvenli şekilde siteye girmemizi sağlar. Sertifikanın süresi dolduğu zaman, siteye girerken güvenlik hatası verir.

wget -i url.txt → url.txt dosyası içindeki tüm url leri indirir

wget -c url → indirme yaparken bağlantı kesilse de bağlantı geldiğinde indirme devam eder

wget --server-response url → sunucu ile ilgili bilgileri alır ve girilen url yi indirir

youtube-dl -F youtubelinki → girilen youtube linkini hangi formatta indirilebileceğini gösterir

format numarası **youtube-dl -f formatno youtubelinki** girilirse; o formatta indirmiş oluruz.
mehmet@ubuntu:~\$ youtube-dl -f 249 https://www.youtube.com/watch?v=ru0K8uYEZWw

youtube-dl --extract-audio --audio-format mp3 youtubelinki → girilen youtube linkini mp3 ses dosyası olarak indirir

convert -resize 300 resim.jpg yeniresim.jpg → yolu girilen resmin boyutlarını 300 300 olarak değiştirir

convert resim.jpg -resize 100x300! yeniresim.jpg → resmin boyutunu 100x300 olarak değiştirir

Örnek : wget --server-response komutu çalışınca gözüken Location kısmının ekrana yazdırılması;
location : kısaltılmış url nin nereye gittiğini gösterir

mehmet@ubuntu:~/linux\$ nano asli

#!/bin/bash

wget --server-response \$1 2>&1 | grep "^Location"

[2>&1 → birinci değişkeni 2 defa kontrol etmesini sağlar]

mehmet@ubuntu:~/linux\$ bash asli istanbulhs.org
Location: https://istanbulhs.org/ [following]
Location: /wiki/ [following]

BİR SORU

Touch a

chmod 07444 a

ls -al a =?

-r - S r - S r -T → ilk üç ve ikinci üç ; kullanıcı ve grup için okuma ve dosyayı root yetkisinde çalıştırma izni vardır. Son üç; T parametresi: Ancak dosyayı ancak oluşturan kişi silebilir.

Apache-Php-Mysql Kurulumu Ve Örnek Bir Uygulamanın Koşturulması

mehmet@ubuntu:~\$ sudo apt install mysql → **mysql'i indirir ve kurar**

mehmet@ubuntu:~\$ sudo apt install mysql-server → **mysql-server indirilir ve indirme sırasında çıkan ekranda mysql-server root şifresi girilir ve kurulum tamamlanır**

mehmet@ubuntu:~\$ mysql_secure_installation → **mysql-server ile ilgili kurulum bilgilerini yapar**

daha sonra /var/www/html/ dizini içerisine info.php dosyası oluşturuyoruz. Ardından içine aşağıdaki kodları yazıp kaydediyoruz.

mehmet@ubuntu:/var/www/html\$ sudo nano info.php

```
<?php
phpinfo();
?>
```

daha sonra web tarayıcımıza localhost/wp yazdığımızda bizden veritabanı, kullanıcı adı vb. Bilgiler istediği için aşağıdaki işlemleri terminalden yapıyoruz;

mysql -u root -p → **mysql komut satırını açar**

mysql> CREATE DATABASE mehmet; → **mehmet isimli veritabanı oluşturur**

mysql> CREATE USER [mehmet@localhost](#); → [mehmet@localhost](#) adında kullanıcı oluşturur

not : mysql de çift tırnak “ yoktur hep tek tırnak ‘ kullanılır.

mysql> SET PASSWORD FOR mehmet@localhost= PASSWORD('PAROLANIZ'); → **kullanıcı için parola oluşturuyoruz**

mysql> GRANT ALL PRIVILEGES ON mehmet.* TO → **mehmet veritabanına ayrıcalık verir**
-> mehmet@localhost IDENTIFIED BY 'PAROLANIZ';

mysql> FLUSH PRIVILEGES; → **tüm ayrıcalıkları kaydediyoruz**

bu işlemi yaptıktan sonra **wordpress.org** sitesinden **wordpress.tar.gz** dosyasını indiriyoruz. Arşivlenmiş bu dosyayı, /var/www/html/wp/ **dizini** altına çıkarıyoruz. Daha sonra bu dizin içindeki **wp-config-sample.php** dosyasını kopyalayarak **wp-config.php** adında yeni bir dosya oluşturuyoruz, **sudo nano wp-config.php** içerisine girerek; aşağıdaki koyu alanlara (girilmeden önce boş olan) bilgileri giriyoruz...

```
.....
/** MySQL database username */
define('DB_USER', 'mehmet');
/** MySQL database password */
define('DB_PASSWORD', 'PAROLANIZ');
/** MySQL hostname */
define('DB_HOST', 'localhost');
```

.....
bu bilgileri girip dosyayı kaydediyoruz. Daha sonra tarayıcımızdan **localhost/wp** sayfasına girip, orada bulunan işlemleri gerçekleştiriyoruz. Sitemiz bu sayede oluşmuş oluyor.