**Linux Çalışma Notları**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ssh {user}@{host}** | | SSK anahtar komutu sisteme şifrelenmiş Secure Shell Bağlantısı açmak istediğinizin komutunu verir. {user} ulaşmak istediğiniz hesabı temsil eder. Örneğin, sistem üzerindeki herhangi bir şeyi düzenlemeye tüm yetkisi olan **root** kullanıcısına erişmek isteyebilirsiniz. **{host}** erişmek istediğiniz bilgisayarı simgeler. Bu bir IP (ör: 244.235.23.19) veya domain (ör: www.xyzdomain.com) adresi olabilir. | | | | | | |
| **Pwd** | | pwd komutu linux işletim sisteminde çalışmakta olduğunuz dizini gösteren komuttur. Açılımı “**P**rint **W**orking **D**irectory” dir yani çalışma dizini yazdır anlamındadır. | | | | | | |
| **Whoami** | | Örneğin; User kullanıcısı root içinde [**su**](https://wiki.ubuntu-tr.net/index.php?title=Su) olarak giriş yaptıysa, whoami rootu gösterirken **echo $USER** ise Useri gösterir. Çünkü su komutu ön tanımlı olarak bir giriş kabuğu yürütmez. | | | | | | |
| **sudo passwd root** | | Root kullanıcısının parolasını oluşturur | | | | | | |
| **sudo passwd -dl root** | | Root hesabını tekrar kilitle. Root hesabını kilitlemek istiyorsan parolayı kaldırıp root’u kilitlemek için aşağıdaki komutu gir: | | | | | | |
| **Su -l** | | Uç birimde root kullanıcısı ile oturum açılır. | | | | | | |
| **Sudo adduser faruk** | | Sadece kullanıcıyı ekler | | | | | | |
| **Sudo useradd faruk** | | Home dizininde kullanıcı dizinini de oluşturup kullanıcıyı ekler | | | | | | |
| **Sudo userdel faruk** | | Faruk kullanıcısı siler | | | | | | |
| **Sudo userdel -r faruk** | | Kullanıcı dosyaları ile beraber kullanıcıyı siler | | | | | | |
| **cd target** | | [Unix](https://tr.wikipedia.org/wiki/Unix) benzeri ve [Linux](https://tr.wikipedia.org/wiki/Linux) [işletim dizgelerinde](https://tr.wikipedia.org/wiki/%C4%B0%C5%9Fletim_sistemi) o an içinde bulunulan dizini değiştirmeye yarayan komuttur. | | | | | | |
| **cd ~(altgr +ü)** | |  | | | | | | |
| **cd /** | |  | | | | | | |
| **Clear (ctrl + l)** | | Terminali temizler | | | | | | |
| Ls -l / (root dizinini ayrıntılı listeler  Ls -lh (Dosyaların Büyüklüklüleriniokunabilir şekilde listeler  ls -lhia : hidden dosyaları da listeler | | Bu komut aktif klasördeki tüm dosya ve klasörleri listeler. | | | | | | |
| ls Belgeler | | Bu komutla aktif klasördeki Belgeler klasörünün içeriği görüntülenir. | | | | | | |
| ls .. | | Bu komutla bir üst klasörün içeriği listelenir. | | | | | | |
| ls -la . | | Bulunduğu dizini gizli dosyalarla birlikte gösterir. | | | | | | |
| ls ../\*.txt | | Bu komutla bir üst klasördeki uzantısı txt olan dosyalar listelenir. | | | | | | |
| ls \*.htm | |  | | | | | | |
| ls ~ | | Bu komutla home dizininizin içeriği görüntülenir. | | | | | | |
| ls ../../Klasor1/Klasor2/deneme.htm | | Bu komutla iki üst klasördeki Klasor1 klasöründeki Klasor2 klasöründeki deneme.htm dosyası listelenir. | | | | | | |
| ls a?e.htm | | Bu komutla aktif klasördeki ilk harfi a olan, üçüncü harfi e olan, uzantısı htm olan, ismi üç harfli olan dosyalar listelenir. | | | | | | |
| ls -l ../deneme | | Bu komut bir üst klasördeki deneme klasörünün içeriğini ayrıntılı olarak listeler. Eğer bir üst klasörünüzde deneme klasörü varsa deneme klasörünüzdeki dosya ve klasörlere bağlı olarak şuna benzer bir çıktı almanız gerekir. | | | | | | |
| \*,- Normal bir dosya  d Dizin  b Özel blok dosyası  c Özel karakter dosyası  l Sembolik bağlantı dosyası  P Özel isimlendirilmiş pipe dosyası | | Bu çıktıdaki ilk karakterler ilgili ögenin dosya mı yoksa klasör mü olduğunu belirtir. İlk karakter d ise ilgili öge klasördür, - ise dosyadır.  ilk karakterden sonra 3 blok halinde gelen soldan sağa doğru user, group ve other olarak sıralayabiliriz. Erişim izinleri herzaman rwx şeklinde verilir eğer rwx yerine “-” işareti varsa o bölüme gelen yerde izin verilmemiş demektir  **s :** suid kim çalıştırırsa çalıştırsın sahibi yetkisinde çalıştırsın  **t:** sticky bit root ve sahibininin silebilmesi ama herkesin dosya açabilmesine izin verir | | | | | | |
| -rwxrwxrwx  chmod u-x soru3.py(User İçin)  chmod g-x soru3.py (group)  chmod o+s soru3.py (other)  chmod 644 soru3.py(-rw-r--r-- | | - Normal bir dosya olduğunu belirtir  r w x r w x r w x 22 +21+20 +22 +21+20 + 22 +21+20+ 7 7 7 | | | | | | |
| Cat dosya1.txt | | | Dosyanın içeriğini standart çıktıya döker | | | | | |
| Cat dosya2.txt | | |  | | | | | |
| cat kaynak.txt > hedef.txt | | | Eğer hedef dosya bulunmuyorsa o zaman komut bu dosyayı yaratacak veya var olan dosyanın üzerine yazacaktır. | | | | | |
| cat kaynak.txt >> hedef.txt | | | Hedef dosyanın içeriklerini de yeni dosyaya eklemek için cat komutuyla birlikte **>>**seçeneğini kullanın | | | | | |
| cat dosya1.txt dosya2.txt > birlesme.txt | | | Bu komut birden çok dosyayı tek bir dosyaya bağlamanıza izin verir | | | | | |
| cat dosya1.txt > dosya1.txt | | |  | | | | | |
| cat dosya1.txt dosya2.txt > dosya1.txt  cat dosya1.txt dosya2.txt > dosya2.txt (hata verir) | | | Sadece dosya2 yi dosya1 e yazar. | | | | | |
| cat -n dosyaadi.txt  cat -v dosyaadi.txt  cat -T dosyaadi.txt  cat -s dosyaadi.txt  cat -b dosyaadi.txt  tac dosyaadi.txt  cat -E dosyaadi.txt | | | 1-Cat komutuyla bir dosyanın içeriklerini her satırın başında rakamlarla görüntüleyebilirsiniz.  2-Bütün bastırılamaz karakterleri görüntülemek için aşağıdaki örnekteki gibi cat komutuyla birlikte **-v** seçeneğini kullanın  3-Yalnızca tab (sekme) karakterlerini görüntülemek için **-T**seçeneğini kullanın  4-Ekranınızda tekrarlanan boş satırları birleştirmek için **-s**seçeneğini kullanabilirsiniz  5-Boş olmayan satırları rakamlandırmak için **-b**seçeneğini kullanın.  6-Bir dosyanın içeriğini son satırla başlayıp ilk satırıyla biten şekilde tersinden görüntülemek için **tac**komutunu kullanın.  Cat komutu satır sonlarını **$**karakterini her satırın sonunda görüntüleyerek işaretleyebilir | | | | | |
| cp \*.sql db\_yedek/ | | | | | | | | Bulunduğumuz dizindeki tüm \*.sql uzantılı dosyaların oluşturulmuş /db\_yedek isimli bir başka klasöre kopyalanma |
| cp second.jpg ../ | | | | | | | | bir üst dizine kopyalar |
| cp second.jpg ../../../tmp/ | | | | | | | | tmp dizinine kopyalamak |
| cp second.jpg /tmp/ | | | | | | | | Tmp klasörüne kopyalar. |
| Cp -avr public ../../public\_backup | | | | | | | | /public adında, içeriğinde dosyalar ve alt klasörler barındıran bir klasörümüzü sahip olduğu yapıyla birlikte başka bir dizine kopyalamak istersek   * -a dosyaya ait file mode, ownership, timestamps ve eğer mümkünse attributes: context, links, xattr verilerini aktarır. * -v işlemle ilgili süreci döker * -R klasör yapısını olduğu gibi kopyalar |
| Cp birinci.txt ikinci.txt | | | | | | | | Birinciyi ikinciye kopyalar. Aynı konumda dosya yoksa oluşturur. |
| man ls | | | | | | | | Man komutu ile komutların help bilgisi gelir. |
| Umask  Umask 0005 dersek değişir | | | | | | | | Henüz oluşturulmuş herhangi bir dosyada hangi izinlerin maskeleneceğini denetleyen sayıdır. 666dan verilecek olan yetki çıkarılır. Makismum 666 ya kadar olan yetkiler verilir. |
| mkdir Linux | | | | | | | | yeni dizin oluşturur ve otomatik gizli 2 dizin oluşur. Bu dizinler aslında işaretçilerdir. |
| rm test (d-base/\* adresini siler.) | | | | | | | | Dosyayı siler |
| rm b\*r | | | | | | | | b ile başlayan r ile biten herşeyi kapsa anlamına gelir |
| rm b?r | | | | | | | | Sadece ortadaki karakter , b ile başlayan ve r ile biter 3 karakterli |
| rm b\* | | | | | | | | b ile başlayan |
| rm \*b | | | | | | | | b ile biten |
| rm \* | | | | | | | | hepsini siler |
| rm -rf \* | | | | | | | | herşeyi alt dizini ile sil ve bana da bir şey sorma |
| rm -r test | | | | | | | | İçindeki herşeyi de siler. Recursive olarak. |
| touch birinci.txt | | | | | | | | touch komutu var olan dosyanın zaman damgasını değiştirir, dosya yoksa yeni bir boş dosya oluşturur |
| mv birinci.txt second.jpg | | | | | | | | Linux'ta dosya ya da klasör taşımak için mv komutu kullanılır. |
| Head /etc/passwd | | | | | | | | İlk 10 satırı gösterir |
| Tail /etc/paswwd | | | | | | | | Sondan 10 satırı gösterir. |
| Tail /etc/passwd -n 3 | | | | | | | | Sondan 3 satır gösterir. |
| tail -n 5 birlestirilmis.txt > dosya2.txt | | | | | | | | 5 satırı eski haline çeviriyor |
| Tail -n +50 dosya adı (50den sonrasını gösterir | | | | | | | | “+satırsayısı” Belli bir satırdan sonrasını göstermek |
| (head b.txt -n 5 && tail a.txt )>yeni.txt | | | | | | | | Parantezin içindeki sonucu yeni txt yazar |
| Ps | | | | | | | | Bu komut verildiği andaki çalışan süreçleri gösterir  PID - Bu benzersiz işlem kimliğidir. Prosess id  TTY - Bu kullanıcının giriş yapmış olduğu terminaldir.  TIME - İşlemin çalıştığı dakika ve saniye cinsinden süre  CMD - Süreci başlatan komut |
| Ps aux | | | | | | | | Bütün kullanıcıların süreçlerini prosesslerini gösterir. |
| İlk süreç /sbin/init | | | | | | | | İlk süreçtir. |
| Ps aux > dosya.txt | | | | | | | | Ps auxun çıktısı dosya.txt ye yazdırır. |
| Ps aux >> dosya.txt | | | | | | | | Append şeklinde çalışır. Dosya txt varsa sonuna ekler yoksa sıfırdan oluşturur. |
| Wc | | | | | | | | Word count (kelime sayacı) |
| Wc -l | | | | | | | | Sadece satır sayısını döndürür. |
| Wc -l dosya.txt | | | | | | | | Dosya içindeki satır sayısını gösterir. |
| ps aux | wc -l | | | | | | | | Şuan için çalışan prosess sayısını gösterir. |
| Wc 3.py | | | | | | | | 12 satır  37 kelime sayısı 271 karakter |
| Grep | | | | | | | Bir yazı dosyasında tanımladığınız bir kalıbı arar. Diğer bir deyişle, grep komutu ile bir kelime veya kalıbı arayabilir ve buna sahip olan satır veya satırları bastırabilirsiniz. | |
| grep [seçenekler] kalıp [DOSYA] | | | | | | | * grep – komutun kendisi * [seçenekler] – komut niteleyicileri * kalıp – bulunması gereken arama sorgusu * [DOSYA] – aramayı yaptığınız dosya | |
| -i – arama büyük küçük harfe duyarlı olmayacaktır. Yani eğer “araba” kelimesi için arama yapmak istiyorsanız “ARABA” da aynı sonucu verecektir.  -c – sadece aranan kalıbın bulunduğu satır sayısını gösterecektir | | | | | | | -r – mevcut dizinde yinelemeli aramayı etkinleştirir  -n – satırlar için arama yapın ve aramanıza uyan satırları alın  -v – bu seçenekle aranan kalıba uymayan satırlar gösterilir | |
| grep -v 1 dosya1.txt | | | | | | | içinde 1 olmayanları gösterir | |
| grep nologin /etc/passwd | wc -l | | | | | | | /etc/passwd nologin olanların sayısını gösterir | |
| grep bash /etc/passwd | wc -l | | | | | | | sistemde oturum açabilen kullanıcı sayısını listeler. | |
| cat dosya1.txt dosya2.txt | grep -v 1 > dosya3.txt | | | | | | | bu kod ile dosya 1 ve dosya 2 de içinde 1 geçen satırlar çıkarılarak dosya3 aktarılır | |
| file /dev/zero | | | | | | | Zero dosyasının ne dosyası olduğunu gösterir. | |
| alias sil="rm -rf" | | | | | | | sil adında takma isimle rm -rf kodu oluşturur | |
| ll komudu alias ll="ls -l" gibi çalışmaktadır | | | | | | |  | |
| unalias sil | | | | | | | Oluşturulan kodu siler | |
| vim .bashrc | | | | | | | aliası kalıcı yapmak için .bashrc ye bu alias sil="rm -rf" kod eklenir. | |
| ls /root/ && cd tmp | | | | | | | ilk ls komudu çalıştığında hata vermezse cd komutu çalışır. | |
| .bash\_history | | | | | | | tail .bash\_history o an komut satırına yazılan kodların tutulduğu yer. Terminal kapatılınca içerik silinmez | |
| echo $PATH | | | | | | | bir komut yazıldığı anda sistem tarafından aranacak olan liste görüntülenir. Örnek olarak who komutu /usr/bin/ dizini altındadır ve bu bilgi PATH değişkeninde saklı tutulur | |
| export PATH=$PATH:/usr/net | | | | | | | buna /usr/net patikasini ekle  $ işareti çevresel değişken olduğu anlamına gelir | |
| echo $EV | | | | | | | Olmadığını görürüz | |
| export EV="/tmp" | | | | | | | ile tanımlaması yapılır. | |
| make clean | | | | | | | derlemeleri siler | |
| htop | | | | | | | etkileşimli bir gerçek zamanlı süreç izleme uygulamasıdır | |
| tar -zxvf htop-3.1.1.tar.gz | | | | | | | Tar.gz uzantılı dosyaları açmak için kullanılır. | |
| ./autogen.sh  make  sudo make install | | | | | | | Derleme işlemi yapılır ve derlemeyi yükler. | |
| apt-cache search htop | | | | | | | htop geçen paketleri listeler | |
| apt-cache search python | wc -l | | | | | | | pyhton adıyla geçen paketlerin sayısını veriyor | |
| sudo dpkg -i paket.deb | | | | | | | deb uzantılı dosya kurmak için | |
| sudo apt -f install | | | | | | | paket yöneticisinin kırık durumunu düzeltir. Eksik paketleri bulup yükle | |
| kill -9 17756 zorla öldürmek istiyorsak -9 ekleriz | | | | | | | bir süreci durdurmak için , numara da o sürecin process id si | |
| killall firefox | | | | | | | Tüm firefox süreçlerini sonlandırır. Alt süreçlerle birlikte | |
| pkill mysqld (sürecin adını kullanarak sonlandırma | | | | | | |  | |
| Cat /etc/passwd | | | | | | | kullanıcıların listesi | |
| Cat /etc/shadow | | | | | | | kullanıcıların parolalarının gölgelenmiş hali bulunur.  Sudo grep ubuntu /etc/shadow ile ubuntu kullanıcısını kontrol | |
| df | | | | | | | diskleri gösterir | |
| sudo fdisk /dev/sdb | | | | | | | diskin seçilerek biçimlendirme işlemine hazırlanır | |
| sudo mkfs.ntfs /dev/sdb | | | | | | | biçimlendirme işlemi yapar | |
| mount /dev/fd0 /mnt/floppy | | | | | | | bağlanmak, takmak anlamındadır. Sistemin yüklü olduğu dosya sisteminin dışında ikinci bir disk, CDROM, disket vs.  (a second hard-drive) ne olursa olsun, bu parçaya erişmeden önce, sistemde var olan bir dizine iliştirilmesi gerekir.  Bu iliştirmeye bölüm eşleme denir. Aygıtın bağlandığı dizine ise bağlanma noktası veya "mount noktası" denir. | |
| Df | disk alanı kullanım bilgilerini görüntüleyen komuttur. | | | | | | | |
| Fdisk -l | sisteminizdeki tüm mevcut disk bölümlerini listeler | | | | | | | |
| Fdisk -l /dev/sda | Belirli bir sabit diskin tüm bölümlerini görüntülemek için aygıt adıyla ' **-l** ' seçeneğini kullanın | | | | | | | |
| sudo fdisk /dev/sdb | Disk seçilerek tüm işlemler listelenir. Buradan işlemler belirtilerek biçimlendirme vs gibi işlemler yapılır.  Önce parttion lar d komutu ile silinir. Daha sonra n komutu ile yeni parttion oluşturulur. W komutu ile kaydedilerek çıkılır | | | | | | | |
| sudo mkfs.ntfs /dev/sdb1  sudo mkfs.ntfs /dev/sdb2 | Üst komutta işlem bittikten sonra bu işlemi geçerli kılmak için biçimlendirme işlemi yapılır. | | | | | | | |
| free | ram ve swap alanını gösterir | | | | | | | |
| swapon --show | Swap detayları görüntülenir. | | | | | | | |
| dd | dosyaları bir formattan diğerine kopyalamak ve dönüştürmek için dd yapıldı. Dd komutu, bir dosyadan bitleri alır ve bunları başka bir dosyaya taşır. | | | | | | | |
| sudo dd if=/dev/zero of=/tmp/swapfile bs=1024 count=1M | zero dosyasını swapfile olarak kopyalar | | | | | | | |
| sudo chmod 600 /swapfile | Oluşturulan swapfile isimli dosyaya yetki verilir. -rw------- | | | | | | | |
| Sudo mkswap /tmp/swapfile | swap dosyası olarak biçimlendirir | | | | | | | |
| sudo swapon /tmp/swapfile | swap olarak ayarlar ama kalıcı hale gelmemiştir. | | | | | | | |
| sudo nano /etc/fstab | işlemi kalıcı hale getirmek için aç en alt kısmına **/tmp/swapfile swap swap defaults 0 0 kopyala** | | | | | | | |
| [**GPL**](https://wmaraci.com/nedir/gpl) (General Public License) | | | | | | Türkçesiyle Genel Kamu Lisansı özgür yazılımlarda kullanılması amacıyla geliştirilmiş bir [yazılım](https://wmaraci.com/nedir/yazilim) lisansıdır. İlk defa 1989’[da](https://wmaraci.com/nedir/da) Richard Stallman tarafından hazırlanan [lisans](https://wmaraci.com/nedir/lisans) daha sonrasında Özgür Yazılım Vakfı tarafından revize edilerek geliştirilmiş ve son haline kavuşmuştur.   * Programın sınırsız olarak kullanma özgürlüğü * Programın nasıl çalıştığını anlama ve özel amaçlara göre değiştirilebilmesi özgürlüğü * Programın kopyalarının sınırsız olarak dağıtılması özgürlüğü * Programın değiştirilmiş halini dağıtma özgürlüğü * **GPL'nin varyantlarından herhangi birini kullanan sistemi veya uygulamayı kapatmamıza izin vermez ve her zaman kaynak kodunu dağıtmamız gerekir**. | | |
| BSD(Berkeley Software Distribution) | | | | | | * Hiç kimse hiçbir kuruluş BSD’nin sahibi değildir. * BSD lisansı yazılım üzerine sınırlama getirmez. * BSD lisansıyla üretilen yazılımların giderleri kamu fonlarından karşılanır. * BSD lisansıyla üretilen yazılımların giderleri kamu fonlarından karşılandığı için yazılımların lisansları da kamu yani halka ait olmaktadır. * İsteyen herkes BSD lisanslı ürünleri kullanma hakkına sahiptir. * BSD lisanslı bir yazılımın kodlarında değişiklik yaparak ve kodları şifreleyerek ya da kapatarak yazılım piyasasında satışa sunabilme imkanına sahip olunabilir * **Öte yandan BSD, kodu görmemize ve değiştirmemize izin verir, ancak aynı zamanda sistemi veya uygulamayı kapatmamıza da izin verir.** | | |
| Shareware | | | | | | * Belirli bir süre **(15 gün, 30 gün vb.)** kullanım izni olan programlara **shareware** adı verilir. Bunların bir bölümü bu süre bitiminde kullanılamaz duruma gelir. Bazıları ise kullanılabilmekte ancak bazı özellikleri işlevsizleşmektedir. * Shareware ile çok benzer olan, ancak kaydetme gibi yaşamsal özellikleri kısıtlanmış programlara ise genelde **demo** adı verilmektedir. | | |
| Freeware | | | | | | * Kullanıcının belirli bir süre sonra yazılımı kullanmaya devam etmek için ücret ödemek durumunda olduğu sharewareden farklı olarak tamamen **ücretsizdir**. Terimin kabul edilmiş açık bir tanımı yoktur. * Bu tür yazılım lisansları, yazılımın bireysel kullanıcıların ücretsiz olarak kullanılmasına imkan verirken, ticari kullanım için üreticisinden izin istenilmesi ve ücret ödenmesi gibi hükümler barındırabilir. | | |
| Uname komutu  -a, --all aşağıdaki leri sırayla ekrana yazdırır  -s, --kernel-name çekirdek adını yazdırır.  -n, --nodename Bilgisayarınızın, ağ içinde kullanılan host adını gösterir.  -r, --kernel-release Kullandımda olan Kernel ana dağıtım bilgisini gösterir.  -v, --kernel-version Kullanımda olan Kernel sürümünün dağıtıma özel sürüm bilgisini, yayınlandığı tarihle birlikte gösterir  -m, --machine Kullandığınız bilgisayarın donanım adını gösterir.  -p, --processor Kullanılan işlemcinin Mimari bilgisini gösterir  -i, --hardware-platform Kullanılan bilgisayarın donanım platform bilgisini gösterir.  -o, --operating-system İşletim sisteminin ne olduğunu gösterir | | | | | | | | |
| **Disk isimleri neye göre isimlendiriliyor** ? örneğin sda1 den neden sda5e geçiyor sorusunun cevabı  MBR bölüm tablosu olan bir diskte, 1'den 4'e kadar olan bölüm numaraları, bölüm tablosundaki "birincil" bölümler olarak adlandırılan dört yuvaya karşılık gelir. Sıralı olarak doldurulmaları gerekmez, bu nedenle örneğin bir sda2'ye sahip olmak ancak sda1 olmaması mümkündür. Bölüm numaraları 5 ve üstü, genişletilmiş bir bölümdeki "mantıksal sürücülere" karşılık gelir ve bunlar her zaman sıralı olarak numaralandırılır, bu nedenle sda5 olmadan bir sda6'ya sahip olamazsınız. | | | | | | | | |
| yazılımın kaynak kodu var. **Nedir kaynak kodu?**  KAYNAK KODU:Kaynak kodu, herhangi bir yazılımın işlenip makine diline çevrilmeden önce insanların okuyup üzerinde çalışabildiği programlama diliyle yazılmış halidir. | | | | | | | | |
| Python,C,Javada vs yazılan kod derlenip bin dosyalara çevriliyor.  1-)Unix benzeri işletim sistemlerinde içeriği akgarıp import ederek kitaplıkları kullanıyoruz.Yazılımlar hep böyle kitaplıklar kullanıyorlar.  Bunların sistmde nasıl barındırılacagı kısmı iki şekilde oluyor  Unıxte her kitaplık sistemde uygun bir yere koyuluyor ve ona ihtiysacı olan yazılımlar gidip o kitaplığı sistemdeki yerinden alıp kullanıyorlar.**PAYLAŞIMLI KİTAPLIK KULLANIMI.**  Windows IOS ANDROID vs her programın ihtiyac duydugu kitaplıkları kendi yanında tasıması seklındedır.  Unixte paylaşımlı kitaplık olayı tek bir merkezden guncellenme durumu oluyor.  Windowsta vs.ise hepsinin ayrı ayrı yazılımların guncellenmesı gerekır.  Unixte bazen programların bazıları 2 bazıları 3. sürümü kullanabılıyor o zaman biz asıl kitaplığı 3 yaparsak sıkıntı olabılıyor.  Avantaj olarkak kurdugumuz yazılımlar cok kucuk disk alanı kaplayacaklar.Bizde bir derleyici mb ile belirtilen disk alanları kaplar iken  her yazılım kendi kitaplığını getirsin diyenlerde gb ile kaplar. Fakat onlar disk en ucuz şey çok sıkıntı yok modundalar. | | | | | | | | |
| Sıkıştırma algoritması(gzip,bzip) tek core kullanır paralel çalışmaz.  **Neden bzip2 paralel çalışmıyor?**  -paralelde çalışır olan kor kadar işi corelara böler fakat sistemin hepsini kullanması hep işimize gwlen bir durum değildir. | | | | | | | | |
| **DOSYA SIKIŞTIRMAK NEDİR? NASIL YAPILIR?**  -Bir veya birden fazla dosyayı kullanılan algoritmayla birlikte daha küçük boyuta indirgeme işlemine sıkıştırma denir.  Sıkıştırma sayesinde dosyaların disk üzerinde kapladığı boyutu düşürülür. Olduğundan daha küçük boyuta indirgenen dosyalar daha kolay indirilebilir,  paylaşılabilir ve gönderilebilir hale gelirler. Böylece kullanıcılar hem zamandan hem internet kaynaklarından hem de depolama alanından tasarruf edebilirler.  Sıkıştırmayı iki farklı şekilde açıklayabiliriz.  Kayıplı sıkıştırma: Kayıplı sıkıştırma yöntemlerinin tümünde orijinal dosya üzerinde değişiklikler meydana getirilir.  Örneğin bir görsel dosyası kayıplı olarak sıkıştırıldığında  boyutunda büyük bir küçülme meydana gelmesine karşın görüntünün kalitesinde de aynı oranda bozulmalar veya netlik kaybı yaşanır.  Kayıpsız sıkıştırma: Bu sıkıştırma biçimindeyse dosyaların orijinali üzerinde kalite açısından bir değişiklik meydana gelmez.  Şimdi masaüstünde yeni bir klasör oluşturup, içerisine bir metin belgesi yerleştirip sıkıştırılırsa bu arşiv dosyası açıldığında da aynı dosyalar aynı kalitede dışarıya çıkacaktır. | | | | | | | | |
| Sistemde her **kullanıcının rootmuş gibi çalıştırabileceği** programların bir listesi bulunur bunlar etc'nin altındadır. **sudoers** altında.  cat /etc/sudoers ile içeriğini görmeye çalışalım göremem. Çünkü dosya okuma iznim yok.  su - yazarız o zaman ve root kullanıcısna geçiş yaparız.  vim /etc/sudoers/ ile root kullanıcısı gibi çalıştırabilecek kullanıcıları belirlemek mümkün olur. ALL yerine komutun kendisini yazmak daha doğru olur. | | | | | | | | |
| neden **su** değil de **su -** ile kullanıyorum örneğin su necdet kullanıcısına böyle de geçiş yapabilirim.  sadece su roota geçer. su - roota geç ve pathinni de al demek istiyorum.PATH her kullanıcı için farklı olan bir çevresel değişkendir. | | | | | | | | |
| **sudo apt update** yapınca bazı sitelere bakıyor.Peki nereye bakması gerektiği bilgisini nereden alıyor?  Unix tarzı işletim sistemlerinde bu tip bilgiler bin dosyalarında değil yapılandırma dosyalarında bulunurlar hemen her insanın okuyabileceği şekildedir.  Hangi depoları kullanmak istediğim bilgisi de yapılandırma dosyaları ile ilgili bilgiler etc dizini altında ve ls /etc/apt/vim /etc/apt/souces.list | | | | | | | | |
| **sudo apt remove apache2**  **fakat yaptığımız değişiklik bile duruyor paketi kaldırmamıza rağmen.Neden?** Normalde gitmesini beklemeli miyim? HAYIR.  \*yapılandırma dosyalarını da yani bir paketle gelen dosyaları da kaldırmak istersek apt remove --purge apache2 ile kullanabiliriz.  yaptığımız değişiklik kaldırılmamıştır çünkü gelen hali değil.zaten paket yöneticisinin işi de değildir bu.  \*apt autoremove  Sistemime kurulu bir takım paketler var bağımlılık olarak kurulmuş ve şimdi ihtiyac yok bunu kaldırır. | | | | | | | | |
| Bilgisayarı açtınız düğmesine bastınız işletim sistemini seçtiniz ve yazılar akmaya başladı yazılar çok hızlı geçiyor.  dmesg yazan yazıları mesajları gösterir.  **dmesg | wc-l** | | | | | | | | |
| **cd /boot/:** Bu dizin bizim işletim sistemimizin açılırken kullandığı dizi ve dosyaları barındırır. | | | | | | | | |
| **Linux çekirdeği** birkaç farklı çekirdek mimarisi var. Monolitik çekirdek tipinde bir mimariye sahiptir. Bir diğeri de microkernel. Bu ikisinin birlikte kullanıldığı hibrit çözümlerde var.  -Linux monolitik bir çekirdek. Bunun anlamı hafızada korumalı mode yüklenmiş bir çekirdek var ve bu çekirdek gerektiğinde diskten modülleri de okuyabiliyor. | | | | | | | | |
| uzantılar:  .csv : comma seperated value  .zip : zip compressed file  .exe : executable file  .apk : android package file  .jar : java archive file  .bin : binary file  .bak : batch file | | | | | dizin ve dosya yapıları  linux tek köklü bir ağaç yapısına sahiptir.(root)  C:\, D:\ gibi gösterimler fiziksel olarak bir diski temsil eder. | | | |
| **vmlinuz** : pcyi açtığımızda çekirdeği boot eden dosyalar. (ls -l /boot) | | | | | **/etc** dizini : yapılandırma dosyalarının barındırır. | | | |
| Gnu/linuxta 1 numaralı süreç sbin/init | | | | | **init** süreci her şeyi boot eden ilk süreçtir. pid'i 1'dir. | | | |
| **ncurses** -> terminaldeki renkli fontları yazmayı sağlayan tooldur. | | | | |  | | | |
| **du -sh file :** file'ın ne kadar yer kapladığını bu komutla öğreniriz. | | | | | **sudo apt dist-upgrade** : yazılım güncelleme | | | |
| **Soft link** (sembolik link) özellikleri şunlardır:   * Farklı dosya sistemleri birbirine linklenebilir (ntfs dosya sistemindeki bir konumu Ext3 dosya sistemindeki bir konuma linkleyebilirsiniz) * Dosya ve klasör için link oluşturulabilir. * Farklı inode sayısı ve yetkilere sahiptir. (inode nedir?) * Orjinal dosyanın yetkileri değiştiğinde linklenmiş dosyanın yetkileri değişmez. * Sadece orjinal dosya veya klasörün konumunu gösterir, içeriği kopyalanmaz. | | | | | **Hard link** özellikleri şunlardır:   * Farklı dosya sistemleri birbirine linklenemez. Aynı dosya sisteminde olmalıdır. * Sadece dosyalar linklenebilir, dizinler linklenemez. * Orjinal dosyayla aynı inode sayısına ve yetkilere sahiptir. * Orjinal dosyanın yetkileri değiştiğinde linklenmiş dosyanın yetkileri de değişir. * Orjinal dosyayla aynı içeriğe sahiptir. | | | |
| **Soft link (sembolik link olarak da geçer)** bir dosya veya dizinin linkidir. **Hard link** ise orjinal dosyanın kopyasıdır. Orjinal dosyayı sildiğinizde **soft link (sembolik link)** mevcut olmayan bir dosya veya dizini göstermiş olur yani boşa çıkar. Çünkü gösterilen yani linklenen dosya veya dizin silinmiştir. Ama hard link tamamen farklıdır. Hard link yaptığınızda orjinal dosyayı içeriğiyle beraber kopyalamış olduğunuz için orjinal dosyayı silseniz bile linklenmiş dosyanın içeriği silinmez. | | | | | | | | |
| Bir dosyanın 5-20 arasını göstermek | | | | | head -n 19 state.txt | tail -14 | | | |
| **Cd** Kabuk isteminde çalıştırabileceğiniz tüm yardımcı programların dosya sisteminde gerçek yürütülebilir dosyalar olarak var olması gerekmez. Bunlar ayrıca kabuk yerleşikleri olarak da adlandırılabilirler.  kabuğun yükleyeceği ve yürüteceği harici bir yürütülebilir [program](https://en.wikipedia.org/wiki/Computer_program) yerine, doğrudan kabuğun kendisinde yürütülen bir [kabuktan](https://en.wikipedia.org/wiki/Shell_(computing)) çağrılan bir [komut](https://en.wikipedia.org/wiki/Command_(computing)) veya [işlevdir](https://en.wikipedia.org/wiki/Subroutine)  Type -a cd çıktısı cd is Shell builtindir | | | | | | | | |
| **RAID ve LVM** arasındaki tek fark, LVM'nin RAID'in sağladığı yedeklilik veya eşlik için herhangi bir seçenek sunmamasıdır | | | | | **RAID ve LVM** ,Bu ikisi arasındaki fark verilerin saklanma şeklidir. | | | |
| **LVM** (Mantıksal Birim Yöneticisi): Disk sürücülerini mantıksal birimlere yapılandırmak için kullanılan bir yazılım aracıdır. Bu araçları kullanarak (donanım destekli) disk sürücüleri, RAID birimlerine ve farklı boyuttaki mantıksal birimlere yapılandırılabilir. | | | | | **RAID** (Yedekli Bağımsız Diskler Dizisi): Veri fazlalığı ve/veya daha iyi performans için depolama disklerini mantıksal birimlerde düzenleme tekniklerinden biridir. | | | |
| RAID, LVM'ye benzemez. Hataya dayanıklılık bölümleri oluşturmak için RAID kullanabilirsiniz, ancak LVM disk bölümleme ve dosya sistemi basımını kolaylaştırmak için kullanılır | | | | | RAID, sürücü yedeklemesi için kullanılır; LVM, RAID cihazınızı mantıksal hacimlere ayırmak için kullanılabilir | | | |
| **Raid Yapılandırması**  Sudo fdisk /dev/sdb (İki bölümün tipi de Linux Raid yapılıyor)  Sudo fdisk /dev/sdc  Sudo mdadm –examine /dev/sd[b-c] diskleri kontrol eder  Sudo mdadm –examine /dev/sd[b-c]1 Disk bölümlerini kontrol eder. MD superblock olmaması gerekli  Sudo mdadm -C /dev/md0 -l raid0 -n 2 /dev/sd[b-c]1 disk bölümlerni alarak verilen raid türünde disk sayısı belirterek Raid oluşturur  Sudo mdadm –detail /dev/md0 verilen blok aygıtının raid bölümlerini gösterir  Sudo mkfs.ext4 /dev/md0 blok aygıtına bir dosya sistemi ve süper blok ataması yapar  Mount /dev/md0 /mnt/raid0/ yaptığımız raidi bir klasöre bağladık. Raid yapısı sistem kapandığında bozulur | | | | | | | | |
| 0 girdi  1 standart çıktı  2 hata çıktısı | | | |  | | | | |
| Make 2> hata.txt | | | | Make komutunda oluşabilecek hata çıktısını hata.txt ye kaydeder  Hata oluşmazsa çıktı olmaz | | | | |
| Make > hata.txt 2>&1 | | | | Make komutunda hata olursa hatayı hata oluşmazsa standart çıktıyı yazar. | | | | |