LINUX TEMELLERİ

**DİZİN KOMUTLARI**

**cd** – dizin değiştirir

**mv** – bir dosya ya da klasörü taşır / ismini değiştirir

**pwd** – print working directory (şuan üzerinde çalışılan dizini ekrana yazdırır)

**ls** – dizin içeriğini yazdırır

* + --all / -a (dizin içeriğindeki gizli dosyaları da gösterir)
  + --listformat / -l (liste formatında göstermek için)
  + --humanReadable / -h (listedeki dosya ve klasörlerin boyutlarını insanın okuyabileceği tarzda (gigabyte falan) gösterir.)

**mkdir** – dizin oluşturur

* + --parent / -p (tam dizin ağacını oluşturmak için)

**rmdir** – boş bir dizini kaldırır

* + --parent / -p (tam dizin ağacını kaldırmak için)

**DOSYA KOMUTLARI**

**touch** – dosya oluşturur

* touch komutunu kullanarak birden fazla dosyayı aynı anda oluşturabilirsin

**rm** – dosyayı kaldırır

* --interactive / -i (aynı anda birden fazla dosyayı kaldırmak için)
* --recursive / -r (özyinelemeli olarak bir klasörü ve içindekileri kaldırır)
* --force / -f (bir dosya yerine bir klasörün kaldırılması hususunda zorlamak için)

**file** – bir dosyanın içerik türü (ASCII metin vs. gibi) ve uzantı bilgilerini gösterir

**cp** – dosyayı kopyalar

* --recursive / -r (bir klasörü kopyalar, başka deyişle özyinelemeli olarak bir dosya kümesini kopyalar)

**echo** – Kendisinde standart girdi olarak (klavyeden) girilen girdiyi, standart bir çıktı olarak (terminal ekranı gibi bir ekrana) yansıtır.

**DOSYA İÇERİK KOMUTLARI**

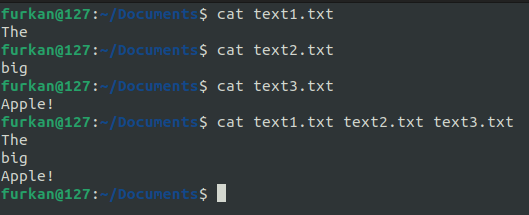
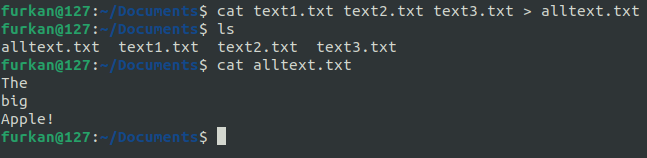
**head** - dosyanın ilk 10 satırı gibi bir parçasını terminal ekranına yansıtır.

* -<satır sayısı> (ilk kaç satır gösterileceğini belirlemek için) -- örnek: head -5 file  file dosyasının içeriğinin ilk 5 satırını gösterir.

**tail** - dosyanın son 10 satırı gibi bir parçasını terminal ekranına yansıtır.

* -<satır sayısı> (son kaç satır gösterileceğini belirlemek için) -- örnek: head -5 file  file dosyasının içeriğinin son 5 satırını gösterir.

**cat -** dosyaların içeriklerinin tamamını gösterir

* cat komutunu kullanarak, birden fazla dosyanın içeriğini bir ekrana yazdırmada birleştirip gösterebilirsiniz:
* Hatta bir yeni bir dosyada da birleştirebilirsiniz:
* Eğer cat komutundan sonra büyüktür işareti (>) koyarsanız, cat komutu bu sefer bir dosya oluşturur ve ardından dosyanın içeriğini yazmanız için bir imleç yanıp söner. İçerik yazma işlemini tamamlamak için ctrl+d basmak yeterli!

Örnek: cat > file.txt  file.txt adında bir dosya oluşturur.

**more –** Terminal ekranına sığmayacak büyüklükte içeriği olan dosyaların içeriklerini sayfa sayfa göstermek için kullanılır. Bir sonraki sayfaya geçmek için ‘Space’ tuşu kullanılır.

**less –** more komutu için daha fazla özelliğe ve artıya sahip olan bir alternatiftir!

**LINUX DOSYA YAPISI**

**Dosya Sistemi:** İşletim sisteminin verilerinin nasıl saklandığını ve alındığını kontrol etmek için kullandığı bir veri yapısı ve yöntemidir.

Veriler, dosya denilen kavramlar içerisinde saklanır. Dosyalar ise dizin denilen farklı türde dosyaların içerisinde barındığı dosya kümeleri içerisinde saklanır. Ayrıca zaten her dizin de esasında birer dosyadır!

**Dosya Sistemi Formatı (Biçimi):** Adı üstünde bir dosya sistemindeki dosyaların diziliş biçimi / yapısıdır.

Mesela Linux İşletim Sistemi Kerneline baktığımızda, ağaç yapısında bir diziliş yapısı görürüz. Yani bir kök dizin (root) vardır ve mevcut olan tüm diğer dosyaları kapsar (alt küme mantığı). Başka deyişle, dosya sistemindeki diğer tüm dizinlerin dolayısıyla dosyaların parent ‘ıdır / atasıdır.

(Veri Yapıları dersi öğrenildiğinde buradaki Parent – Child ilişki ile kast edilen daha net anlaşılacaktır!)

**Bir Diski Formatlamak / Bir Bilgisayara Format Atmak:** Bu iki terim size çok tanıdık geliyor değil mi? Günlük hayatta neredeyse her birerimizin başına mutlaka gelen bir hadise! Peki “Ne farkı var bu ‘bir diske format atmak’ kavramının ‘bir diski boşaltmak / bir diskin içerisindeki dosyaları silmek) kavramından?” şeklinde sorduğumuzda; karşımıza şu şekilde kalıplaşmış ama yanlış olan söz çıkar: “İkisi de aynı şey değil mi ya?”

Cevap: Tabii ki de aynı şey değiller! Şöyle ki ‘dosyaları silmek / diskin içini boşaltmak’ dediğimiz şey, adı üstünde sadece belli bir format / dizilişteki dosyalar arasından bazılarını ortadan kaldırmak anlamına gelirken; bir diski biçimlendirmek / formatlamak denildiğinde ise kast edilen şey, o diskin üzerindeki dosyaların diziliş şeklinin değişmesidir. Yani o diskin üzerindeki dosya sisteminin formatının (biçiminin) değişmesidir özetle. Her bir işletim sistemi Kernel’ine özel bir dosya sistemi formatı mevcuttur. Bunlardan biri ise Windows Kerneli’nin dosya sistemi formatı olan ve Windows NT versiyonu için tasarlamış olduğu NTFS (New Technology File System) dir.

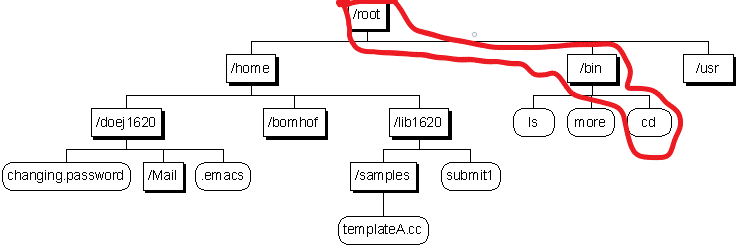
**Dosya Yolu (Path):** Adından da anlaşılacağı üzere bir dosya yolu; bir dosyanın, mesela Linux için ağaç veri yapısı şeklinde modellenmiş olan dosya sistemi üzerinde belirtilen herhangi bir dizine kadar olandosyanın kök dizine kadar giden parentlarının kümesidir.

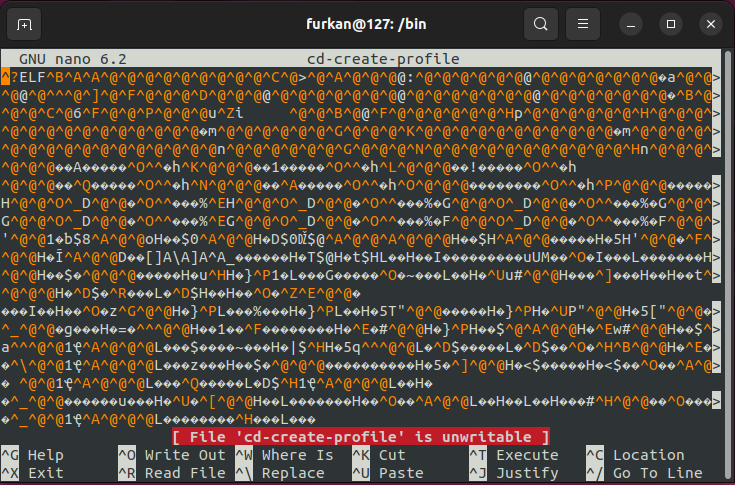
**Kök Dizininin Bir Altında Yer Alan Önemli Dizinler**

**bin: Linux dosya sisteminde bütün binary dosyalarının saklandığı dizindir. Bu binary dosyalardan kast edilen ise linux terminal komutları veya birtakım temel yardımcı programlardır.**

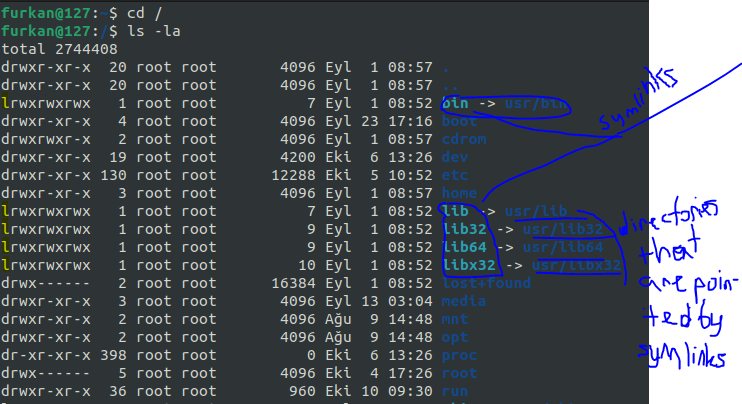
Bilişim dünyasında herhangi bir komut satırında yazılan o komutlar aslında bir kelimeden ibaret değildir. Onlar esasen Shell uzantılı bir dosyadır. Yani sadece Linux Kernelinin anlayacağı dil olan Assembly dilde ve normal bir metin dosyasının içerisine yazılmış talimatlar dizisidir diyebiliriz. Haliyle bu Binary dosyasının da dosya sisteminde bir dosya yolu mevcuttur öyle değil mi? İşte biz komut satırı arayüzüne bir komutu yazdığımızda aslında onun alternatif dosya yolu dediğimiz bir dosya yolu versiyonunu yazarız halbuki asıl yazmamız gereken o komutun temsil ettiği o Binary dosyasının yoludur!

Mesela ‘cd (change directory)’ komutunu ele aldığımızda, bu komutun Linux Dosya Sistemi ağacındaki dosya yolu aşağıdaki görselde verilmiştir:



 Tıpkı ‘cd’ komutu gibi Linux sisteminde yer alan diğer tüm komutlar da aslında o komutların temsil ettiği bir veya birden fazla Binary dosyasının yoludur. Mesela ‘cd’ komutu birden fazla dosyaya sahiptir, bunlardan biri olan ‘cd-create-profile’ dosyasının içeriği aşağıdaki gibi bir insanın okuyabileceği tarzda değildir.

NOT: Komut satırında, root dizininde ‘ls’ ile içerik listelendiğinde, aşağıdaki görseldeki gibi ‘bin’ dizinin başında bir ‘l’ harfinin olduğu görülecektir. Buradaki ‘l’ harfi, ‘link (bağlantı)’ anlamınadır.

Linux işletim sisteminde ‘Symlink (Symbol Link)’ denilen bir kavram mevcuttur. Söz konusu symlinkin amacı adı üstünde bir dizine işaret eden / bir dizin için bir bağlantı görevi gören başka konumda bir ağlantı dizini oluşturmaktır, bu bağlantı dizinine de Symlink adı verilir.

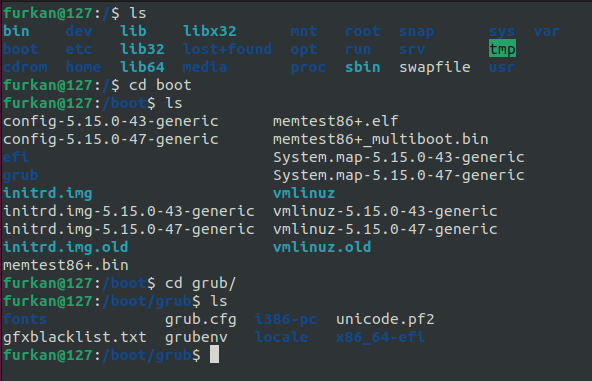
Bir symlinkin başka bir dizine işaret etmesi, görselde de görüldüğü gibi, ‘->’ işareti ile gösterilir!

**sbin: Açılımı: ‘superbin’ olan ve ‘bin’ dizini ile aynı mantıkta işlev gören bu dizin, ‘bin’ den farklı olarak sadece süper kullanıcı / root kullanıcı tarafından yürütülebilen bazı belli başlı komutları içerir.**

**boot: Linux sistemindeki çok kritik önem taşıyan dizinlerden biri olmakla beraber, bünyesinde linux işletim sisteminin ön yükleme işlemi esnasında gerekli olan dosyaları barındırır.**

Çoğu tipik linux kullanıcısı bu dizini, işletim sisteminin kalan kısmından farklı bir disk bölümünde olmak suretiyle ayırır, özellikle de dual booting (çift ön yükleme) yaparken! Ama varsayılan olarak linux için normal kurulum seçildiğinde (manual falan değil), farklı bir disk bölümüne ayrılmadan kurulur.

Boot dizini ayrıca içerisinde grub adında küçük bir program daha barındırır. Bu küçük program, özellikle dual boot yaparken ön yüklemesini yapmak üzere bir işletim sistemi seçmenize ve ön yüklemesini yapmasına olanak tanır. Söz konusu grub programı ile alakalı dosyalar ise boot dizininin altında grub adlı bir dizin altında yer alır:



**cd-rom: Linux sistemi yüklü olan bilgisayara bir cd, dvd bir mount ettiğinizde (bağladığınızda), söz konusu imageların (görüntülerin) bağlama işleminden sonra yer aldıkları dizindir. Başka deyişle, linux sisteminin cd görüntü drive’ının (sürücüsünün) saklandığı dizindir. Siz bir cd / dvd takmadığınız müddetçe dizin boş kalıcaktır.**

**dev: Linux işletim sisteminde hard disklerin, usb belleklerin ve optik sürücülerin (cd / dvd) bağlandığı dizindir. Optik sürücülerin ‘cdrom’ dizini varken ek olarak bu dizin içerisine de bağlanabilmelerinin sebebi ise; dev dizinin, linux’un üzerine inşa edilmiş dağıtım türüne göre optik sürücüler için bir extra depolama alanı ihtiyacı olması ya da söz konusu dağıtımın stillinin o şekilde olmasıdır.**

Eğer bünyesinde birden fazla bölümlendirme barındıran bir sürücü (mesela bir hard disk) bağladıysanız, muhtemelen linux (yine dağıtıma göre değişiklik gösterir) söz konusu disk bölümlerinden bir kısmını ek olarak dev içerisinde saklayacaktır.

**media: Genellikle usb bellekler gibi media amaçlı bilgisayara bağlanan sürücüler için ek bir depo alanı olan ve var olma konusunda Linux’un dağıtımlarından birinden diğerine farklılık gösteren bir dizindir.**

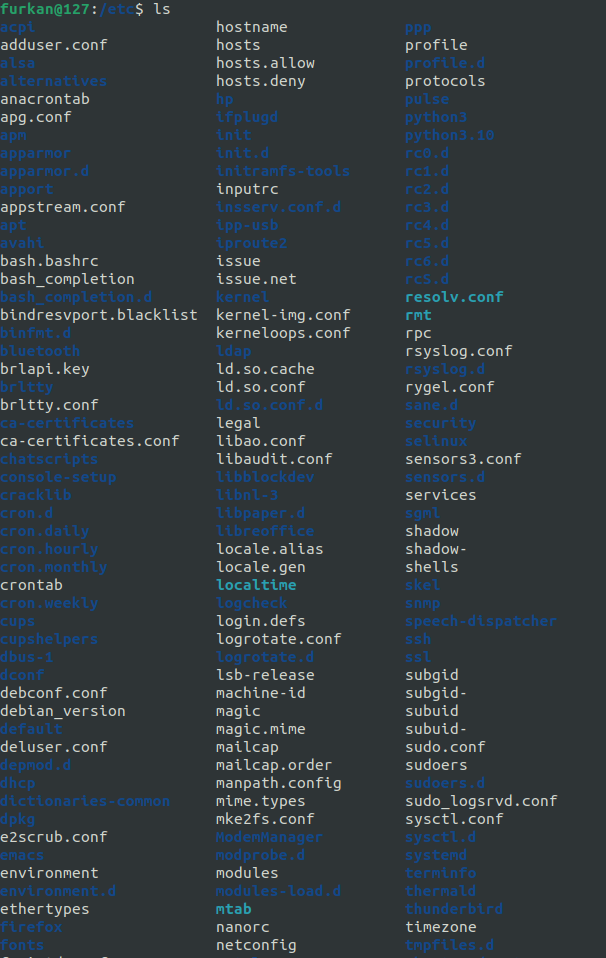
Her Linux dağıtımında olmamakla beraber, harici olarak bilgisayara bağlanan cihazlar (usb, hard disk vs.) için alternatif bir dizin niteliğindedir.

**mnt: Açılımı ‘mount (bilgisayara bir cihazı bağlama)’ olan bu dizin tıpkı media dizini gibi, harici olarak bilgisayara bağlanan cihazlar (usb, hard disk vs.) için alternatif bir dizin niteliğindedir.**

**Note:** media ve mnt dizinleri arasındaki bariz fark: mnt dizini sisteme geçici olarak bir çıkarılabilir sürücü ya da bir sunucu bağlama işlemleri bir bağlama noktası iken; media ise bildiğimiz çıkarılabilir medyalar / sürücüler için bağlama noktasıdır! Mesela Linux sistemine bir sunucu bağlanacağı zaman mnt dizini kullanılır.

**run: Tıpkı mnt dizini gibi, harici olarak bilgisayara bağlanan cihazlar (usb, hard disk vs.) için alternatif bir dizin niteliğindedir. Başka deyişle yukarıda geçen dizinlere bir alternatiftir.**

Ancak bu dizini onlardan ayıran küçük bir ayrıntı fark mevcuttur o da bu dizinin, masaüstü ortam sürücüleri için bir bağlanma noktası olmasıdır. Yani mesela bu dizin kullanılarak, Linux’un masaüstü ortamı olan GNOME modifiye edilebilir ya da başka bir masaüstü ortamı çalıştırılabilir.

**etc: Açılımı bir rivayete göre : ‘editable-text-configurations’, bir diğer rivayete göre ise: ‘etcetera’, yani ‘düzenlenebilir metin yapılandırmaları’ / ‘ve sair ‘ olan bu dizin; bünyesinde, sistem üzerinde built-in (başdan beri var olan) yahut sonradan yüklenmiş tüm uygulamalara ait yapılandırma dosyalarını ve sistem veri tabanlarını barındırır.**

Mesela siz bir Apache sunucusu yüklediğinizde, söz konusu sunucuya ait olan tüm config. (yapılandırma) dosyaları bu dizin altında saklanacaktır.

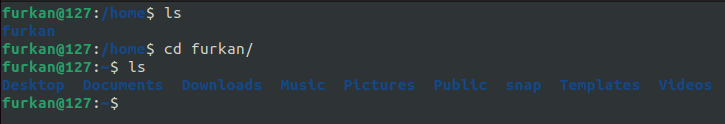
Bunun dışında mesela SSH bağlantınız üzerinde değişiklik yapmak istediğinizde yine bu dizin ile muhatap olacaksınız çünkü SSH protokolü ile alakalı tüm yapılandırma dosyaları bu dizin içerisinde yer alır.

Bu dizin içerisinde yer alan her yapılandırma dosyası, İngilizcesinin kısaltımı olan .conf uzantısı ile muhafaza edilir.

**home: İçerisinde tüm kullanıcı isimli klasör / dizinleri barındıran dizindir.**

Mesela ‘furkan’ isminde bir kullanıcının, işletim sistemi değil de kullanıcı tarafından düzenlenebilir tüm kişisel doküman, video vs. dosyaları ve uygulamaları, home dizininin altındaki söz konusu kullanıcının ismiyle isimlendirilmiş (kullanıcının home dizini olarak nitelendirdiğimiz) dizin içerisinde yer alır**.**

Linux işletim sisteminde bir kullanıcı oluşturulduğu zaman, Linux işletim sistemi anında home dizininin altında kendisi için belirlenen isimde bir home dizini oluşturur.

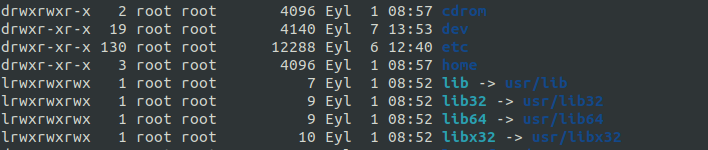


**Not:** Görselde de görüldüğü gibi, bir kullanıcının home dizininin simgesi: ‘~’ işaretidir.

Root dizininin (kök dizin) ise: ‘/’.

**lib: Açılımı ‘library (kütüphane)’ olan ve içerisinde kullanıcının indirmiş olduğu uygulamalara ait, binary dosyaları gibi işletim sistemi tarafından execute etmede kullanılan yapı dosyalarının bulunduğu dizindir.**

Bu dizin için, Windows’taki ‘Program Files’ dizini ile aynı mantıkta ve işlevde olduğu söylenebilir.

Bunun dışında, bu dizinin kendisinden hariç, tıpkı Windows’taki Program Files dizinin 32 ve 64 bit olmak üzere ayrılması ile aynı mantıkta iki farklı tipi daha mevcuttur ki bunlar: ‘lib32’ ve ‘lib64’ dizinleridir. Dolayısıyla indirilmiş bir program / uygulamanın işlemci tarafından 64 bit olarak işlenen program parçaları lib64 içerisinde saklanırken; 32 bitlik parçaları ise lib32 içerisinde yer alır.

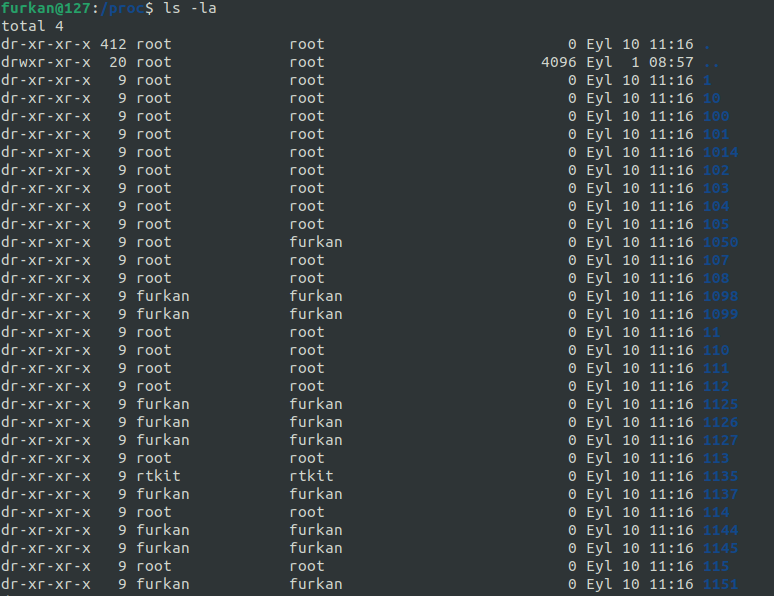
**opt: Açılımı ‘optional (isteğe bağlı)’ olan ve işletim sistemi için isteğe bağlı yazılımların bulunduğu dizindir.**

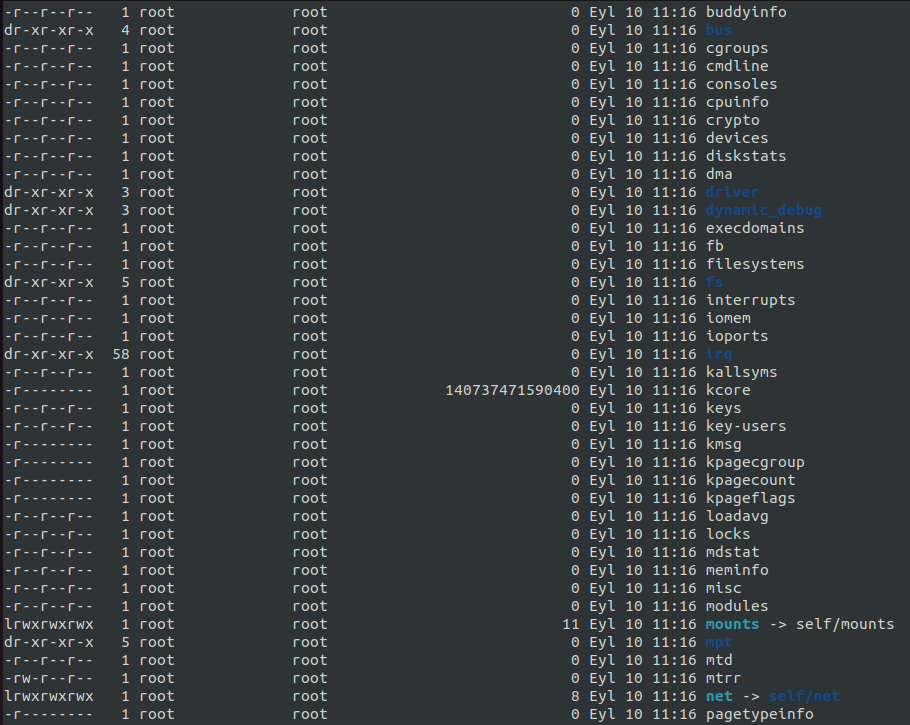
Ne var ki bu dizin, kullanıcı tarafından hiçbir zaman bir uygulama ya da program dizini olarak kullanılmayacaktır. Ancak tabii ki, bu kullanılan Linux Dağıtımına göre değişir.

**proc: Açılımı ‘processes’ olan ve işletim sistemi için kritik öneme sahip olan bu dizin, işlemci taraflı process, işletim sistemi taraflı ise taskların dosyalar halinde muhafaza edildiği yerdir. Başka deyişle ‘Prosesler Klasörü’ denilebilir.**

Görevi ise, Linux Kernelinin, Linux ortamında çalışan taskların çeşitli proseslerinden bilgi alma ve onlara bilgi gönderme işlevini yerine getirmesi için bir depo vazifesi görmekir.

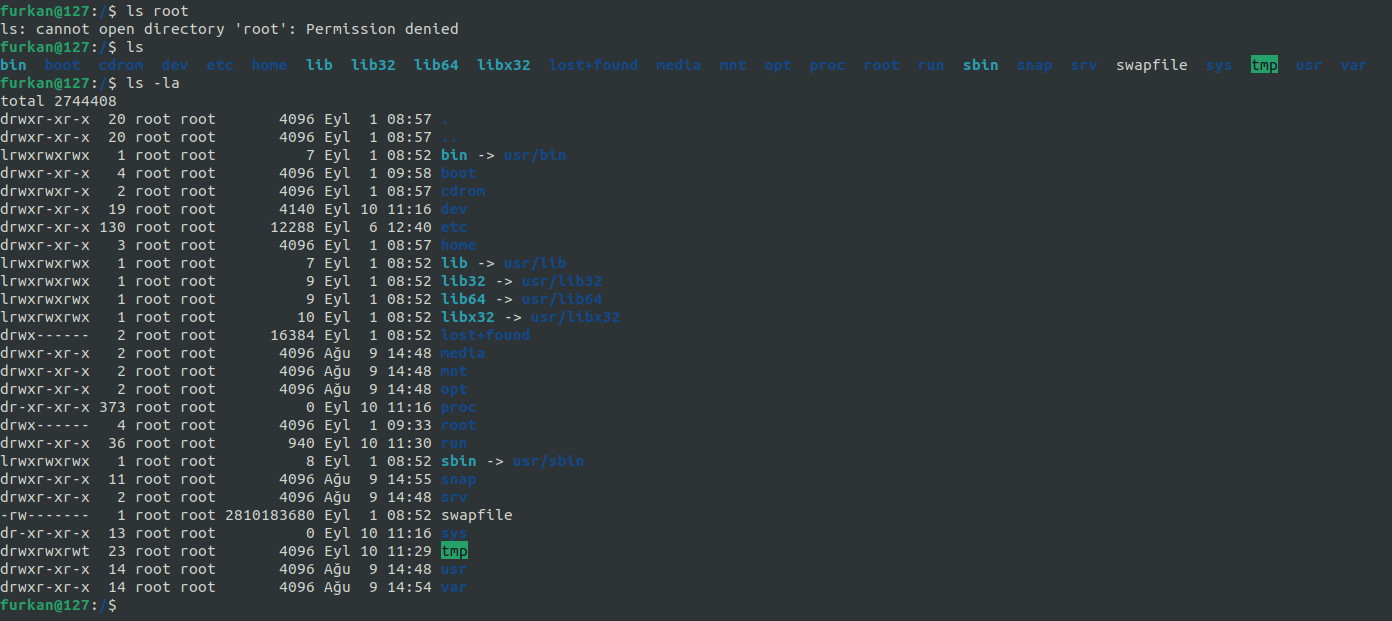
İçerisinde iki tür process dosyası mevcuttur: 1-Process ID ile temsil edilenler, 2-Process Name ile temsil edilenler.

Aşağıdaki ilk görsel 1. Türe, ikincisi ise 2. Türe örnek olan proseslerin bir kesitini içerir:



**root: İçerisinde bulunduğumuz ‘root’ dizininden farklı bir şey olan bu dizin, ‘root’ kullanıcı hesabı için bir home dizinidir.**

Nasıl ‘home’ dininin altında normal bir kullanıcı hesabının home dizini yer alıyorsa, aynı mantıkta ‘root’ dizininin altında da ‘root’ kullanıcı hesabına ait olan home dizini yer alır ve işte o home dizini bu dizindir. Başka deyişle bu dizinin içerisinde, root kullanıcısının kişisel ve hesapla alakalı dosyaları yer almaktadır.



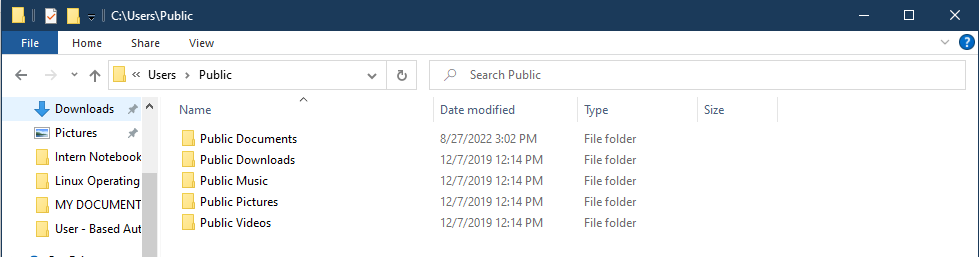
Görüldüğü üzre, sadece root kullanıcısı bu home dizinine erişebilir. Çünkü yetki sadece onda!

**tmp: Açılımı: ‘temporarly (geçici)’ olan bu dizin, içerisinde her işletim sistemi yeniden başlatıldığında silinen geçici dosyaların saklandığı dizindir.**

Başka deyişle, RAM içerisinde yer alan proseslerin fiziksel anlamda tutulduğu dosyaların işletim sistemindeki mantıksal anlamda tutuldukları dosyaları içerir.

**snap: Uygulamaları snap paketleri olarak anılan paketler ile konteynerler halinde sisteme yüklemenin yeni nesil bir yöntemidir ve Ubuntu sistemlerinde çokça görülür. İşte bu dizin, bu tip yükleme işlemine özel bir ortamdır.**

**usr: Kullanıcılar arasında paylaşımlı olarak kullanılan dosyalar ve yararlı programların saklandığı yerdir. Tıpkı Windows işletim sistemindeki Users dizini altında yer alan Public dizini gibi:**



**var: Açılımı: ‘variable (değişken)’ olan ve değişken verilerin tutulduğu dizindir.**

Söz konusu değişken veriler genelde sistem loglarıdır (kayıtlarıdır). Ancak başka türde değişken veri dosyaları da içerir.

Mesela siz bir Apache sunucu ayağa kaldırdığınızda ve o sunucunun varsayılan konumu, bu dizin altında yer alan ve ‘www/html’ olacaktır! Ayrıca söz konusu sunucunun log dosyaları da yine bu dizin içerisinde yer alacaktır.

**swapfile: Fiziksel Bellekte saklanan şeyler için ek bir depolama alanı olan bu dizin, sizin ilk sistem kurulumunda disk bölümlendirmesini yaparken swap alanının oluşturulma seçeneğini seçip seçmemenize göre var olur. Ancak varsayılan olarak sistemde oluşturulacaktır.**

Başka deyişle sistem belleği için mantıksal bir yedek depolama alanıdır diyebiliriz.

**SİSTEM YÖNETİMİ**

**Sistem Bilgi Komutları**

**Bu komutların genel amacı, komut satır arayüzü üzerinden sistem ile alakalı bilgileri elde etmektir.**

**uptime –** Sistemin ne zamandan beri ayakta olduğunu başka deyişle çalışır vaziyette olduğunu ve bunun yanında, bu süre zarfında kaç kullanıcı tarafından kullanıldığını ve işlemci yük ortalamasını gösterir.

NOT: İşlemci yük ortalaması terimini izah etmeden önce bir işlemci yükünün ne demek olduğunu açıklamak daha mantıklı olacaktır:

İşlemci Yükü: CPU tarafından yürütülmekte olan yahut yürütülmeyi bekleyen process / işlem adedidir.

15:23:38 up 13 days, 8 min, 1 user, load average: 3.84, 3.72, 2.41

Mesela yukarıdaki uptime komutu çıktısındaki yük ortalaması değerlerine baktığımızda:

* İlk değer (3.84): son 1 dk içerisindeki,
* İkincisi (3.72): son 5 dk içerisindeki ve
* Üçüncüsü ise (2.41): son 15 dk içerisindeki ortalama işlemci yükünü (yürütülen / bekleyen işlem adedini) temsil eder.

Söz konusu değerlerdeki her bir rakam, yürütülen / bekleyen bir işlemi temsil etmektedir. Ancak bir ayrım vardır o da:

1 rakamına kadar olan sayılar yürütülen bir işlemi, başka deyişle işlemcinin bir çekirdeğinin kullanım yüzdesini ifade ederken; 1’ den sonraki sayılar ise haliyle işlemciyi bekleyen işlemlerin adedini / işlemcinin kullanılmayan kalan kısmını ifade eder!

**ALTERNATİF: Söz konusu değerlerin sayıları, her bir CPU Clock darbesini / CPU zamanını ifade eder.**

Ancak durum, işlemcinin core (çekirdek) adedine göre değişiklik gösterir, zira bir işlemci çekirdeği esasında bir işlemci demektir. Bu yüzden her bir işlemci çekirdeğine bir işlem tahsis edildiğinden; işlemcinin çekirdek adedine göre, yürütülen ve bekleyen işlem adedi değerleri değişiklik gösterecektir. Aşağıdaki örnek, bunun açık bir izahıdır:

Örnek:

23:16:49 up 10:49, 5 user, load average: 1.00, 0.40, 3.35

Yukarıdaki bir linux sistemine ait olan uptime ekran çıktısında yük ortalamasına bakıldığında, sistemin işlemcisi:

* Eğer tek çekirdekli ise:

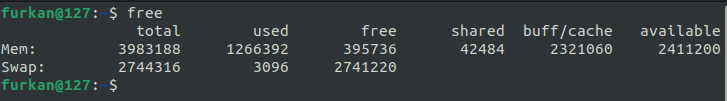
1. Son 1 dk içerisinde tüm işlemci bir işlem için tahsis edilmiş / kullanılıyor demektir. Yani 1 rakamı; bir işlemi temsil ettiği ve bir işlemci çekirdeği de bir işlemci saati periyodu / zamanı başına sadece bir işlem yürütebildiği için, otomatikman **kullanım bakımından** 1 işlemci çekirdeğini temsil eder.
2. Son 5 dk içerisindeki değere bakıldığında; daha 1 rakamına ulaşılmadığı için herhangi bekleyen bir işlemden söz edilemez. Dolayısıyla işlemci, bir işleme tahsis edilmiştir ve o işlem de söz konusu işlemcinin %40 ‘ını kullanıyor, başka deyişle %60’ı kullanılmıyor demektir.
3. İşlemcimiz tek çekirdekli olduğu için sadece 1 tane işlem tahsis edilebileceğinden, 3.35 ‘in 1’i yürütülen işlem ; 2.35’i ise bekleyen işlem adedini ifade etmektedir.

Kullanım yüzdesi olarak ele aldığımızda, işlem kuyruğuna başka işlem eklenmez ise; işlemcinin tamamı (%100’ü) **şu an** kullanılıyor, daha sonra bir daha tamamı kullanılacak ve son olarak %35’i kullanılacak demektir.

* İki çekirdekli ise:

1. Son bir dakika içerisinde; bir çekirdek boşta, diğeri ise kullanılıyor; CPU zamanını bekleyen herhangi bir işlem yok! (1.00)
2. Son 5 dk içerisinde; CPU’nun %160’ı boşta; CPU zamanını bekleyen herhangi bir işlem yok. (0.40)
3. Son 15 dk içerisinde; her iki CPU da kullanılıyor, 1.35 işlem CPU zamanını beklemede. (3.35)

**free –** Sistem belleğinin kullanılan ve serbest ola miktarını görüntüler.

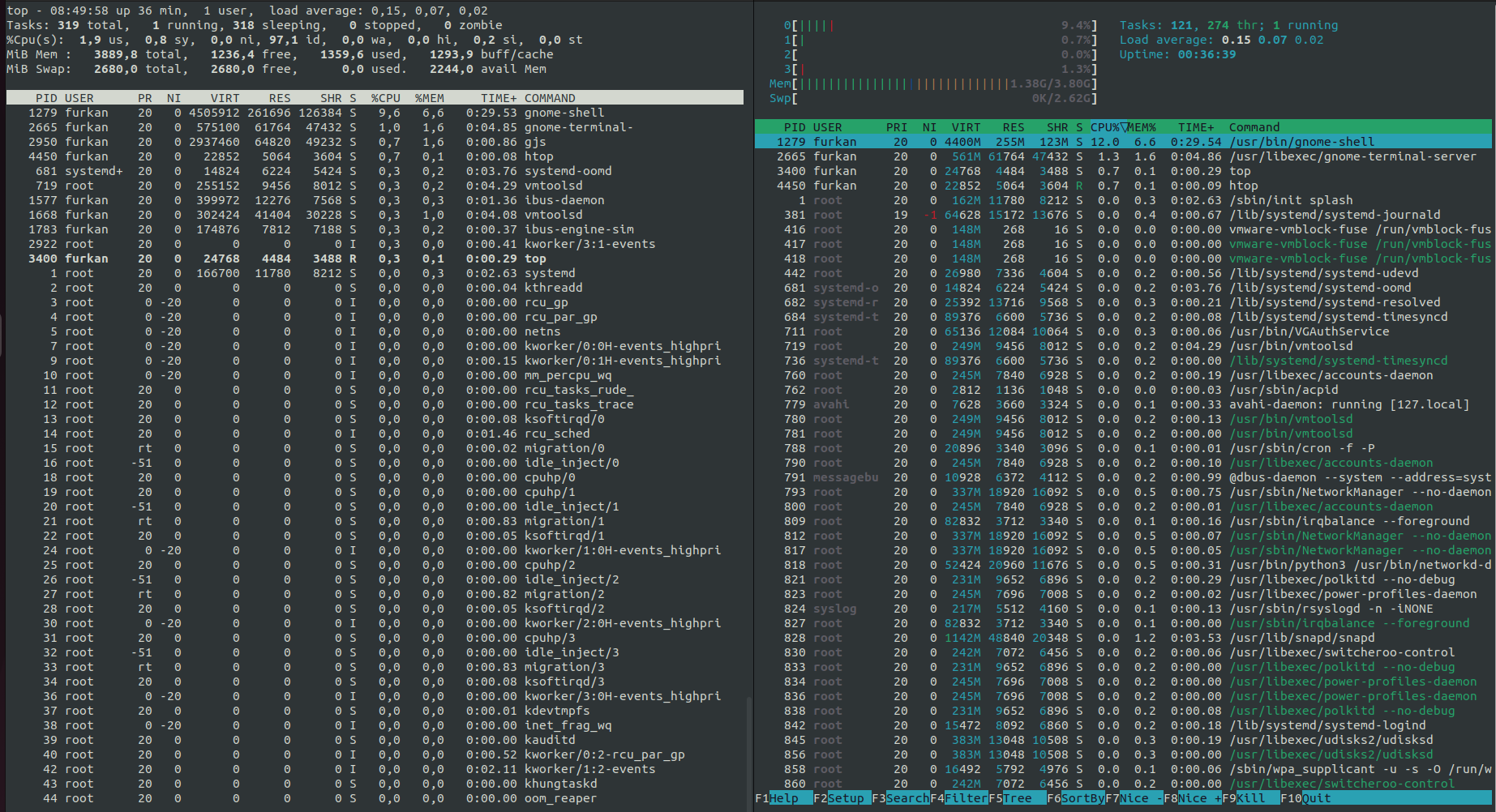


**ps –** Anlık yürütülmekte olan process listesini görüntüler.

İşlemlerin hepsini görebilmek adına -A / --all seçeneği kullanılabilir.

**top –** ps komutunun kısıtlı bir versiyonudur. ps den farklı olarak sadece anlık olarak en üstte çalışan (işlemcide ilk sıralarda olan) prosesleri göz önüne getirir.

Ayrıca top ın özel bir versiyonu olan ve grafiksel anlamda göze daha çok hitap eden bir araç da htop tır. Ancak kendisi build-in olmadığından package manager vasıtasıyla indirmelisiniz.

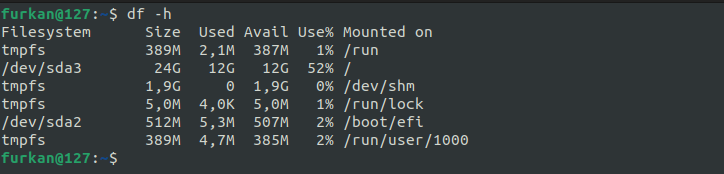


Yukarıdaki sol görseldeki komut çıktısı top a ait iken; sağ taraftaki ise htop a aittir.

**Hard Drive / Disk Komutları**

**df –** İşletim sistemindeki dosya sistemi / disk bölümlerinin disk kullanımını rapor eder .

Bu komutu genellikle –human readable / -h seçeneği ile kullanırız. Böylelikle herhangi bir disk bölümünün giga byte bazında ne kadar kullandığını ve ne kadarın serbest olduğunu rahatça görebiliriz.

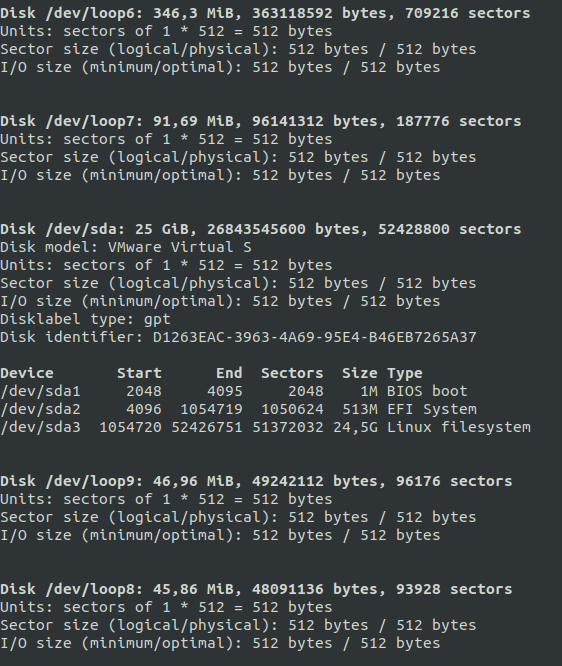
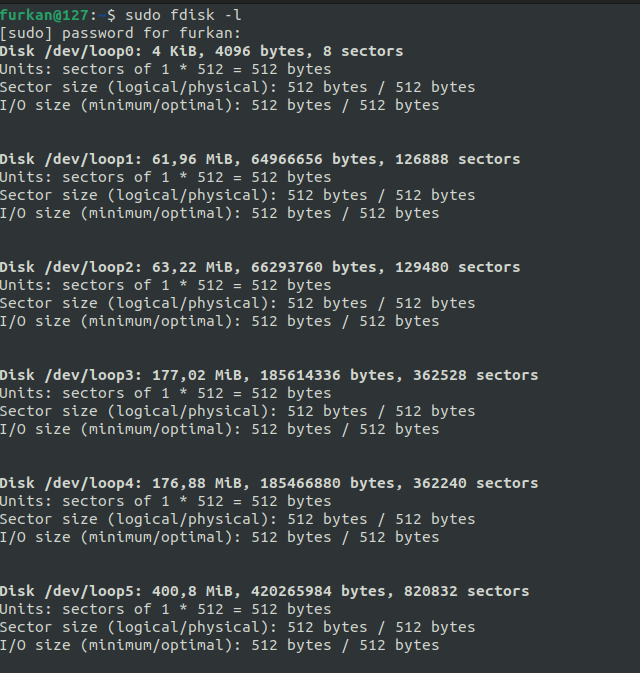
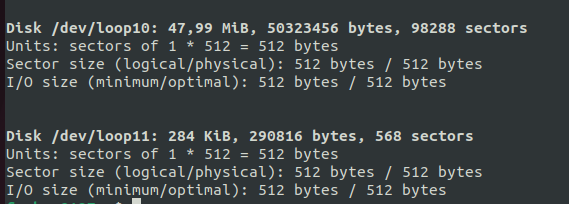


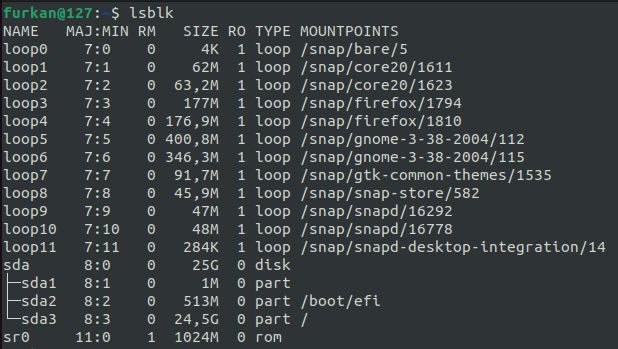
Yukarıdaki görselde;

* **Filesystem:** Dosya sisteminin partitionlarının (bölümlerinin) listelendiği sütundur.
* **Size:** Adı üstünde ilgili dosya sistemi bölümünün diskte kapladığı toplam alanı göstermektedir.
* **Used:** İlgili partitionun ne kadarının kullanıldığını söyleyen sütundur.
* **Avail:** Aslen Available olan bu sütun, ilgili disk partitionının kullanılmayan / serbest olan kısmını göstermektedir.
* **Use%:** Used sütunundaki kullanım oranını % olarak gösterir.
* **Mounted On:** İlgili disk bölümünün dosya sistemin üzerinde hangi dizine bağlandığını / bağlı kalarak idame ettiğini gösteren sütundur. (Mesela yukarıdaki resimdeki sistemde, işletim sisteminin /dev/sda3 üzerinde kurulu olduğunu anlamak çok da zor değildir. Çünkü mount edildiği / bağlandığı yer root dizinidir. Ayrıca en fazla kullanım yüzdesine sahip olan disk bölümü de odur.)

**fdisk –** Disk partition tablolarını manipüle (kendi isteğimiz doğrultusunda düzenleme) etmemizi sağlayan komuttur. Başka deyişle, df komutunun editlenebilir versiyonudur diyebiliriz.

* Eğer fdisk komutundan sonra –list / -l seçeneğini ilave ederseniz, dosya sistemi üzerinde var olan tüm disk bölümlerinin bir listesinin çıktısını önünüze getirecektir. Ancak bu işlem ciddi ve bilgiler son derece mahrem olduğundan tabiiki root yetkisiyle bu komutu çalıştırmalısınız!



**lsblk –** Bu komut, fdisk ve df komutlarının bir adım ötesine giderek dosya sistemindeki bölümler altındaki blok cihazları (Alt cihazları) görüntüler.

**AĞ**

**Temel Ağ Bilgisi Edinme Komut ve Araçları**

**ifconfig** – İşletim sisteminin Ağdaki pozisyonu kapsamında, bilgisayarın ağ katmanı ve aşağısında olan katmanlar ile alakalı temel bilgilerinin çıktısını veren komuttur. İşlev bakımından Windows’taki ipconfig gibidir.

**ip –** ifconfig in yeni ve daha gelişmiş bir versiyonudur. Zira ifconfig ile yapılamayan pek çok şey, şuan ip komutu ile yapılabilmektedir. Mesela ipsec / vpn tünellerini göstermek ya da yönlendirme işlemlerini manipüle etmek gibi.

**netstat –** Açılımı: ‘network state (ağ durumu)’ olan ve networking konusunda çok faydalı komutlar arasında yer alan netstat; TCP bağlantıları, yönlendirme tabloları ve bir dizi ağ arayüzü ve protokolü istatistiğini görüntüleyen bir komut satırı arayüzü aracıdır.

**nslookup –** Domain adı ve IP adresi arasındaki eşleşmeyi elde etmek adına DNS sorgusu yapmak için oluşturulan bir ağ yönetimi komut satırı aracıdır.

**Temel Ağ Yapılandırma Komutları**

**ifconfig –** Bu komut sayesinde temel birkaç ağ yapılandırması yapabilirsiniz. (ip adresleme türünü düzenleme gibi)

**ip –** Bu komutu kullanarak gelişmiş ağ yapılandırmalarını yapmak mümkün. (statik ip, rota vs. vermek, yönlendirme tablosunu düzenlemek, DNS ve ağ geçidi üzerinde değişiklik yapmak ve daha birçok şey…)

**Temel Ağ Erişim Araçları**

**telnet –** Açılımı: ‘terminal over network (ağ üzerinden terminal)’ olan bu araç, size uzaktan ağ üzerinden erişim amaçlı bir terminal ortamı sunar. Bu sayede uzaktaki bir bilgisayarın Shell arayüzünde tıpkı localhost ‘daymışcasına istediğinizi yapabilirsiniz.

**ssh –** Açılımı: ‘secure Shell’ olan bu ağ aracı, güvenli olmayan bir ağ üzerinden iki ana bilgisayar arasında güvenli bir şifreli bir bağlantı sunar. İşlev bakımından telnet ile neredeyse aynıdır fakat güvenlik bakımından aynı şeyi söyleyemeyiz!

**LINUX PAKET YÖNETİCİSİ**

**apt –** Açılımı: ‘advenced package management tool (gelişmiş paket yönetim aracı)’ olan bu araç, temel anlamda linux yazılım paketlerinin yüklenmesi ve kaldırılması gibi işlem ve süreçlerin yapılmasında kullanılır.

Başka deyişle, Debian tabanlı dağıtımlar için bir paket yöneticisi vazifesi görür.

Ayrıca kullandığınız dağıtıma göre, başka dağıtımlara has olan paket yöneticileri de mevcuttur. Ubuntu dağıtımında apt’dir.

Mesela sistemi güncellemek (içerisindeki paketleri, en güncel hallerine getirmek) ve yükseltmek (içerisindeki yükseltmesi olan paketleri yükseltmek) için sırasıyla:

sudo apt update

ve

sudo apt upgrade

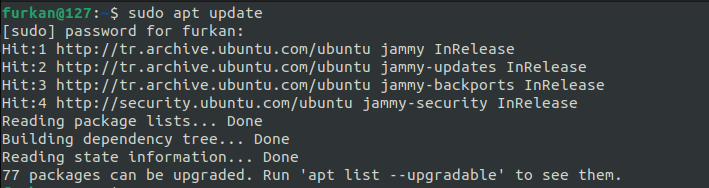
komutları kullanılır. Ancak yukarıda olduğu gibi bu komutlar kök dizini yetkisinde çalıştırılmalı; çünkü bir sistemi güncellemek ve yükseltmek, kullanıcının izni dahilinde olmadan gerçekleşemez! Ayrıca diğer bir sebep ise, söz konusu yazılım paketlerinin işletim sistemine yani root kullanıcıya ait olmasıdır.

**“Peki bu sistemi güncelleme işlemi tam olarak nedir?” sorusunu yönelttiğimizde;**

Çoğu Linux dağıtımı, sisteme yükleyebilmeniz hususunda kendisine yetki verilmiş yazılım paketlerinin içerisinde saklandığı depolara sahiptir.

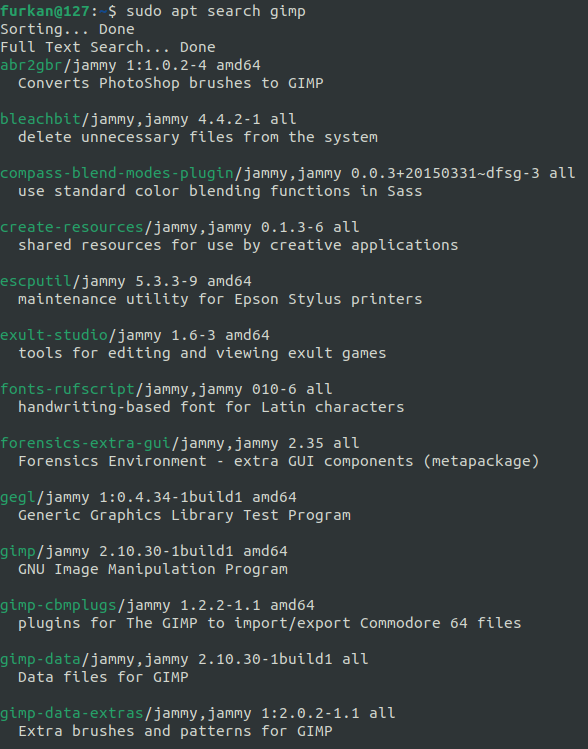
Bir sistemi güncellemek esasında, söz konusu depolardaki yazılım paketlerinin yeni bir sürümü mevcutsa yahut var olmayan bir paket eklenecekse; bu paketlerin temin edilmesi ve deponun paket üreticisinin en son çıkardığı sürüme sahip olunması işlemidir. Bu minvalde, söz konusu güncelleme işleminin devreye girmesi için elbette ki bir komuta / talimata ihtiyaç vardır:

Apt komutu beraberinde gelen bir seçenek olan Update (Güncelle) seçeneği, temelde işletim sisteminiz için kullanılabilir paketler önbelleğini günceller. Böylelikle depoda eğer sisteminize yüklenmiş olan yeni bir paket varsa; söz konusu Update seçeneğini komut satırına yazdıktan sonra bu paketin ne olduğunu görebilirsiniz. Esasen Update seçeneğinin amacı budur.



Update seçeneği tam olarak, yazılım paketlerinin saklandığı tüm ilgili Linux Dağıtımının sunucularındaki depoları gezer ve güncel olan tüm sürümleri veri bazında çekerek işletim sisteminin yukarıda bahsi geçen kullanılabilir paketler önbelleğini bu yeni sürümler ile günceller. İşlem sonunda, yeni bir sürüme yükseltilebilecek olan paket adedini verir ve dilerseniz apt aracının başka hayati önem taşıyan seçeneği olan ‘Upgrade’ ile adedi verilen ve yeni sürüme sahip olan paketlerin yükseltme işlemini başlatabilirsiniz.

Sözün özü, Update seçeneği size yeni sürümlerden haber verirken; Upgrade ise söz konusu yeni sürümleri yükler ve uygular.

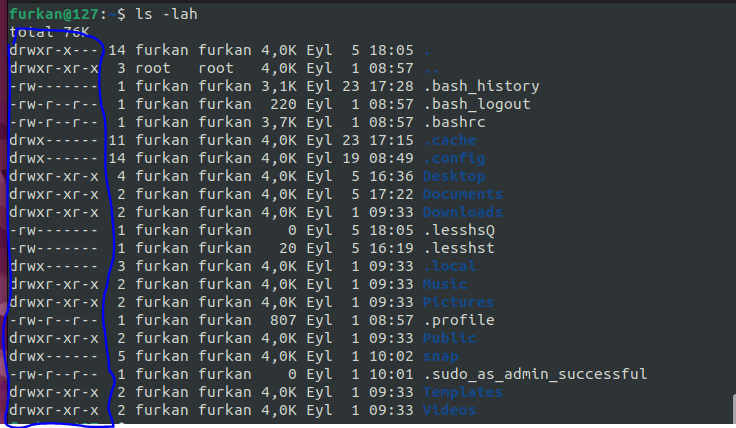
Apt aracının, yukarıdaki iki seçenek dışında ‘search’ diye bir seçeneği mevcuttur ve bu seçenek sayesinde yüklemek veya sistemin deposundaki varlık durumu hakkında bilgi sahibi olmak istediğiniz bir yazılımı aratabilirsiniz. Söz konusu seçenek size ilgili yazılım ile alakalı depoda tutulan tüm paketleri ve dependency dediğimiz bağımlılıkları (söz konusu yazılımın çalışmak / yürütülmek için ihtiyaç duyduğu tüm ek yazılım, kaynak kod vs.) bir çıktı halinde terminal ekranına getirecektir (tabii eğer sistem deposunda öyle bir yazılım varsa, aksi takdirde çıktı vermez!).

Bir yazılımı işletim sistemine yüklemek için ise, apt aracının Install seçeneğini kullanıyoruz.

Kaldırma işlemi için ise Remove!

**LINUX DOSYA VE DİZİN ERİŞİM İZİNLERİ VE SAHİPLİK**

ls komutu ile bir dizinin içeriğini listelediğimizde; listenin ilk sütununda birtakım garip harfler görürüz. Esasında o harflerin her birinin bir anlamı vardır ve Linux işletim sistemindeki ilgili dizin ve dosyalara erişim izinlerini temsil / ifade etmektedirler.



**d:** directory (dizin)

**r:** read (okuma izni)

**w:** write (yazma izni)

**x:** execute (çalıştırma / yürütme izni)

**-:** none (erişim izni yok)

Şöyle ki, (eğer varsa) d harfinin dışında ilk 3 harf, dosyanın yahut dizinin ait olduğu kullanıcının erişim izinlerini temsil ederken; sonraki 3’ü söz konusu kullanıcının üyesi olduğu gruptaki kullanıcıların hepsinin erişim izinlerini ve son 3 harf ise Linux işletim sistemi üzerindeki diğer tüm kullanıcıların erişim izinlerini temsil etmektedir.

Yukarıda da görüldüğü gibi sonraki sütun ilgili dizin ya da dosyanın içerdiği parça adedini temsil eder. Mesela Desktop dizini toplam 4 adet parça içermektedir. Parçadan kastımız ise bir alt dizinde bulunan her dosya / dizindir.

Bir sonraki sütunda ise ilgili dosya / dizinin sahibi olan kullanıcının adı yer almaktadır. Linux’ta her dosya / dizinin sahibi olan bir kullanıcı bulunmaktadır.

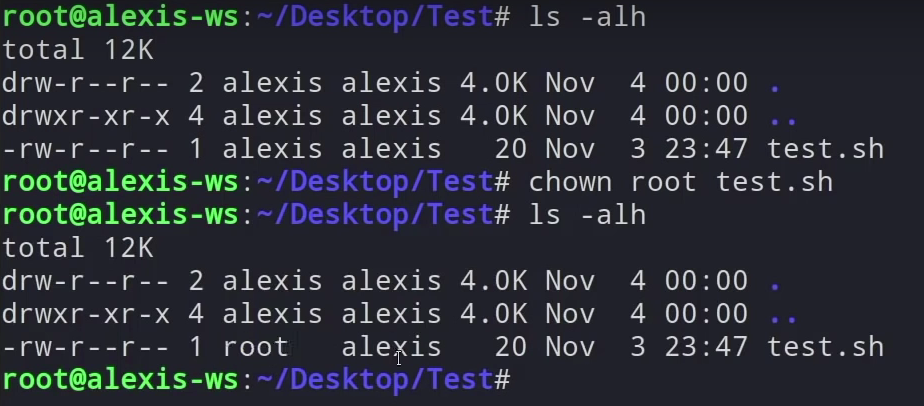
Bir sonraki sütunda ise ilgili dosya / dizinin sahibi olan grubun adı (kullanıcı grubu) yer almaktadır.

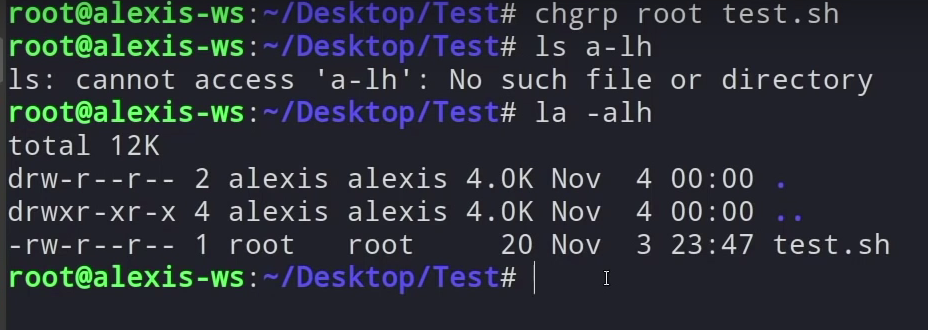
Sonraki sütunların da zaten sırasıyla neleri ifade ettiklerini biliyoruz.

**Bir Dosya / Dizinin Sahibini Değiştirmek**

Linux’ta bir dosyanın / dizinin sahibi olan kullanıcıyı değiştirmek için: “chown”,

“ “ “ “ “ “ grubu değiştirmek için ise: “chgrp”

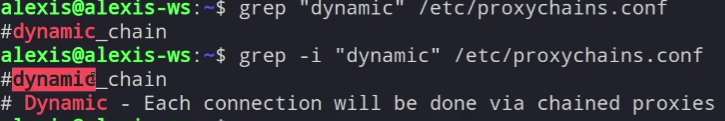
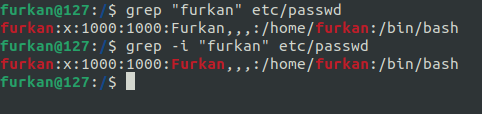




**BİR DOSYA İÇERİSİNDE BİR KELİME ARATMAK**

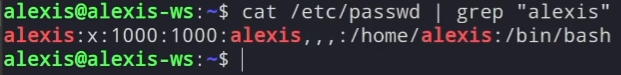
* Linux komut satırında bu eylemin alternatif birçok yöntemi olmasının yanında; bunlardan birisi de **‘grep’** komutudur.

Basit manada **grep (global regular expression print: evrensel düzenli ifade yazdırımı)**, dosyaları bir harf katarı (string) için tarayan ve ilgili katara rast geldiği satırları ekrana yazdıran küçük bir komutlar ailesidir.

Normalde grep komutu büyük-küçük harf duyarlılığına sahiptir. Ancak eğer yanında **--ignore-case** **/ -i** seçeneğini kullanacak olursanız, bu sefer duyarlılığı aldırmaksızın arama yapacaktır.

* Söz konusu yöntemlerden bir diğeri ise ‘|’ işareti ve ‘cat’ komutu kombinesidir.

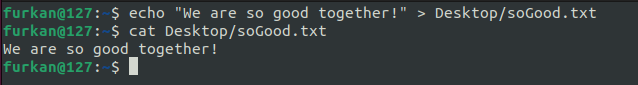
Şöyle ki mantıken yapılan şey tam olarak cat komutu ile ekrana yansıtılacak olan dosya içeriğinin tamamını, pipe işareti vasıtasıyla grep komutuna redirect etmektir (yönlendirmektir). Sonrasında ise yönlendirilen bu içerik içerisinde, grep komutu ilgili kelimeyi aratıp ekrana yansıtacaktır.

Aslında normal cat’in kullanımından tek fark; cat komutu tüm dosya içeriğini ekrana çıktı olarak yansıtmak yerine sadece pipe vasıtasıyla grep’in arayıp da bulmuş olduğu satır üzerindeki kelimeler ekrana yansıtılmış olur. Yani bir nevi cat komutu için istisnai bir durumdur diyebiliriz.

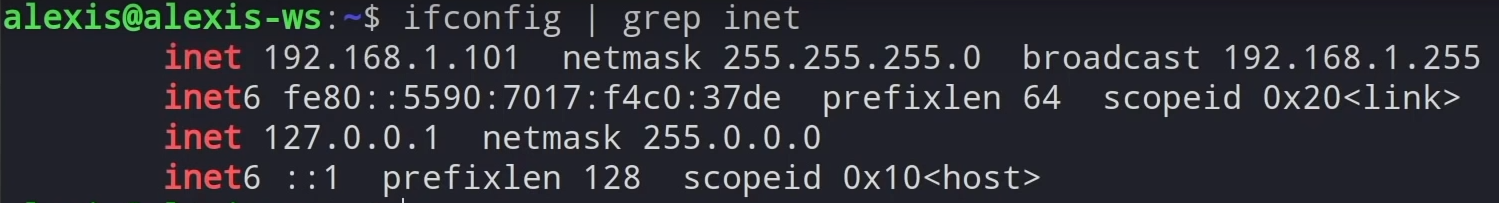
**NOT: PİPE İLE REDİRECT İŞARETLERİNİN FARKI**

Büyüktür işareti olarak da isimlendirdiğimiz yönlendirme işareti (‘>’), bir çıktıyı bir dosyaya gönderme veya bir komuta standart bir girdi olarak bir dosyayı okuma işlevleri için kullanılırken; pipe ise standart bir çıktıyı, başka bir komuta standart bir girdi olarak geçirme işlevi için kullanılır.

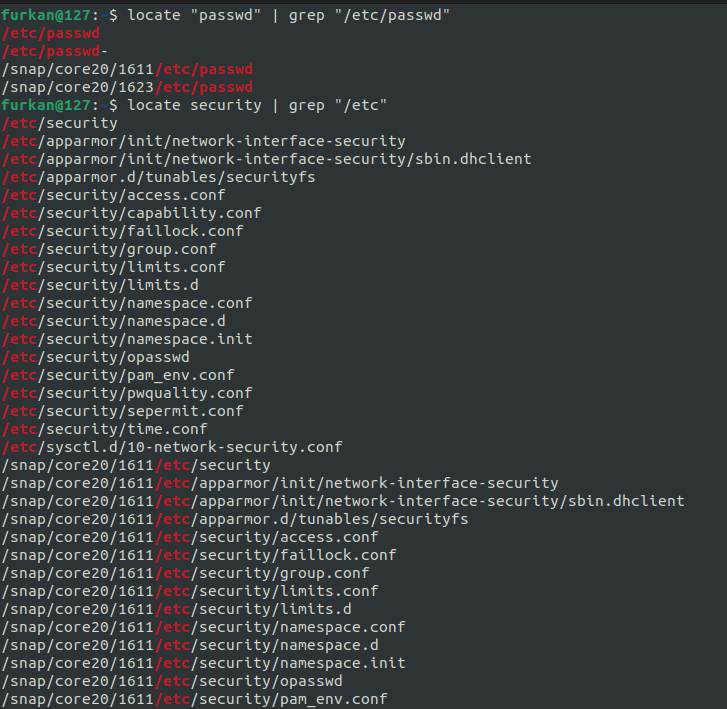
Yukarıdaki işlevleri görseller ile izah edecek olursak;

1. 
2. ./myprogram < inputfile.txt

Bu ifade, programınızı (myprogram.sh) yürütür ve inputfile.txt içerisindeki verileri programınıza girdi olarak pompalar.

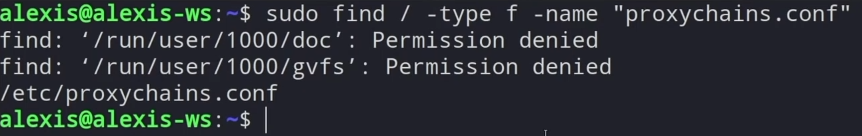
1. 

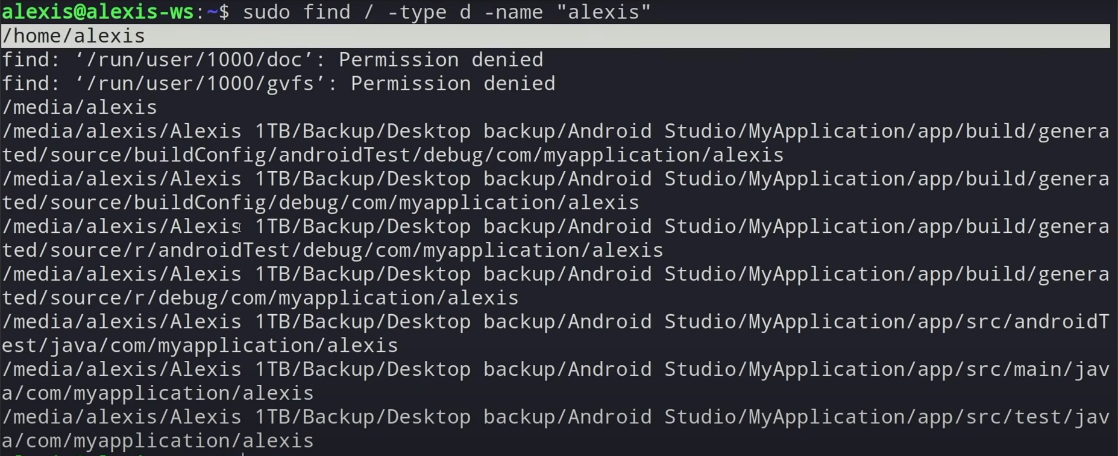
**LINUX SİSTEMİNDE DOSYA BAZINDA BİR KELİME ARATMAK**

**locate –** İşlev bakımından grep komutu ile aynıdır, bariz fark: Grep dosya içeriğinde tarama yaparken, locate Linux sisteminde dosyalar arasında tarama yapar. Yani anlayacağınız biri dosya içeriğinde satır bazlı arama yaparken, diğeri dosya bazlı arama yapar.

**BİR DİZİN İÇERİSİNDE BİR DOSYA / DİZİN ARATMAK**

**find –** Bir dizin hiyerarşisindeki dosyalar için arama yapar.

Genel Formül: find + <aranacak dizin adı> + ‘-type’ seçeneği + <tip adı (dosya/dizin)> + ‘-name’ + <dosyanın / dizinin ismi>



**İÇERİSİNDE BULUNULAN LINUX DAĞITIMI VE KERNEL HAKKINDA BİLGİ EDİNME KOMUTLAR LİSTESİ**

**Aşağıda listelenen komutların tamamının ne olduklarını ve ne işe yaradıklarını öğrenmek için ‘whatis’ komutunu kullanmak, eğer o da yetmezse ilgili komutun manual sayfasına göz atmak yeterlidir.**

KULLANICILAR HAKKINDA

whoami

hostname – değiştirmek için ise: sudo nano /etc/hostname

id

groups <kullanıcı adı>

İŞLETİM SİSTEMİ HAKKINDA

lsb\_release -a / cat /etc/issue

cat /etc/os-release

cat /etc/\*release

cat /etc/redhat-release

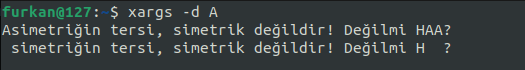
DONANIM HAKKINDA

lscpu

uname

**TERMİNALE HERHANGİ BİR ŞEKİLDE ARGÜMAN OLARAK GİRİLEN BİR GİRDİ ÜZERİNDE OYNAMA YAPMAK**

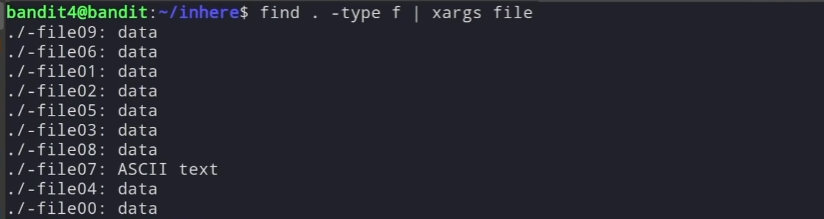
Başlıkta bahsettiğimiz durumu sağlayabilmenin bir yolu XARGS (EXECUTE ARGUMENTS – Argümanları Yürüt) komutudur. Şöyle ki bu komut, diğer Linux Komut Satırı komutlarından farklı olarak; kendisine girilen argüman / argümanla üzerinde (argümandan kasıt: mesela cd komutu için bir dizin yahut file komutu için bir dosya vs.) farklı kısıtlama, filtreleme, düzenlemeler yaparak çıktıyı farklı kılabilir. Mesela aşağıdaki resimde xargs komutu, bir seçeneği olan -d / --delimited (sınırlı) ile kullanılmıştır. Bu seçenek, xargs komutunun çıktısında ne gibi bir farklılık oluşturacak derseniz; siz söz konusu seçeneği belirtikten sonra hangi karakteri girerseniz, standart girdi olarak kullanıcıdan alacağı argüman metnindeki o harflerin hepsini silip o şekilde çıktının oluşmasını saplayacaktır:



Genel Formül: xargs [options] [command [initial-arguments]]

**BİR DİZİNDEKİ TÜM DOSYALARIN İÇERİKLERİNİN NE TÜR OLDUKLARINI GÖREBİLMEK**

Başlıkta bahsedilen durum, XARGS komutu ile sağlanabilir. Şöyle ki xargs komutuna standart bir input vermek yerine pipe yardımıyla eğer bir dosya listesi verirsek; otomatikman kendisine argüman vericilik hizmetinde bulunduğu komut da o girdiyi kendisine argüman kabul edecektir. Mesela biz file komutunu sadece kendi başına kullanırsak, argüman: standart bir girdi (kalvyeden kullanıcı tarafından girilen) olacağı için sadece o girdi için vazifesini yerine getirecek dolayısıyla da sadece bir dosyanın türünü gösterebilecektir. Fakat eğer ki biz file komutuna argüman olarak xargs ve yardımıyla bir liste (pipe vasıtasıyla yönlendirilen bir komutun çıktısı) verebilirsek; o zaman listedeki tüm dosyaların türleri ekranımıza gelecektir. Gelin bu anlatılanları bir de görselde inceleyelim:

Görsele baktığımızda, mesela ‘/home/furkan/.local/share/session\_migration-ubuntu’ dosyasının içeriğinin ASCII Metin formatında olduğunu, başka deyişle insan okuyabilir bir formatta olduğunu görebilmekteyiz.

**LİNUX SİSTEMİNDE SIKIŞTIRMA & ARŞİVLEME**

İngilizceleri sırasıyla ‘to compress’ ve ‘to archive’ olan bu iki terim kullanım bakımından bir bütün olan iki ayrı işlemdir. İlk olarak sıkıştırma işlemini ele alalım;

SIKIŞTIRMA

Şöyle ki sıkıştırma dediğimiz şey: Linux sistemi üzerindeki herhangi bir (özellikle metin formatında olan) dosyanın içeriğinde birtakım algoritmalar aracılığı ile belli başlı değişiklikler yapmak suretiyle, söz konusu dosyadan veri kaybı söz konusu olarak ya da olmayarak dosyanın boyutunun azaltılması işlemidir. Yani esasen bizim sıkıştırma işlemini kullanmamızdaki ana gaye, dosya sisteminde yer alan ve boyut anlamında sıkıntı veren bir dosyanın, daha küçük bir boyuta indirgenip o şekilde idame etmesini sağlamak!

Bir önceki paragraftan da anlaşılacağı üzere, iki tür sıkıştırma vardır: 1- Kayıplı (Daha az önemli kabul edilen verilerin azaltılma anında kaybı söz konusudur.), 2- Kayıpsız (Herhangi bir veri kaybı yaşanmaksızın orijinal verinin, çıktı olan sıkıştırılmış dosyada tekrar oluşturulması söz konusudur.)

Kayıplı sıkıştırmada dosyanın boyutu önemli ölçüde aza indirgenebilir ama dosyanın kalitesi (veri bazında) daha düşük iken; kayıpsızda ise herhangi bir veri kaybı söz konusu değildir ve bunun bedeli olarak da tabii ki boyutu, kayıplıya göre bir tık daha yüksek olacaktır.

SIKIŞTIMA İŞLEMİNE TAHSİS EDİLMİŞ DOSYA UZANTILARI

Bir dosya sıkıştırıldığında doğal olarak çıktı olan dosyanın uzantısı, girdi olan dosya ile aynı olmamalıdır. Bu yüzden Linux geliştiricileri, tıpkı diğer ‘.txt, .c, .sh …’ gibi sıkıştırma işlemi için belli başlı uzantılar tahsis etmişlerdir ve işin güzel tarafı da bu uzantılar, söz konusu sıkıştırma işlemini Shell ortamında gerçekleştiren komutların kısaltımlarıdır!

Aşağıda sıkıştırma komutları anlatılırken söz konusu uzantılara da değinilecektir.

SIKIŞTIMA İŞLEMİNE TAHSİS EDİLMİŞ KOMUTLAR

NOT: ‘ZIP’ sözcüğünün herhangi bir açılımı yoktur. Ancak Bilişim dünyasında ‘sıkıştırma’ eylemini refere etmektedir.

En bilindik olan ve arşivleme için oldukça kullanılan bir komut, ‘gzip’ komutudur.

**gzip –** Temelde sıkıştırma algoritması olan ve açılımı ‘GNU zip’ olan bu komut, Linux Dağıtımlarında sıkıştırma işlemini gerçekleştiren komutların başında gelmektedir.

gzip ile sıkıştırılan bir dosya, ‘.gz (z: zip, g: gnu)’ uzantısına sahiptir ve 10 byte lık bir başlık, isteğe bağlı ekstra başlıklar, bir checksum dediğimiz (bütünlük kontrolü) ve orijinal sıkıştırılmış dosya boyutunu gösteren veriler içerir.

**gunzip –** gzip ‘in “unzip / uncompress (skıştırılmış bir dosyayı orijinal haline geri getirme)” versiyonudur. Yani gzip ile sıkıştırılan dosyalar gunzip ile eski hallerine getirilebilirler (tabii başka unzip komutları da mevcuttur!).

NOT: gzip aracının -d / -decompress seçeneği, adı üstünde gzip’in tıpkı gunzip gibi sıkıştırılmış bir dosyayı genişletmeye yarar. Buradan şu denklem elde edilir:

gzip -d / --decompress = gunzip

Bir başka önemli algoritmayı kullanan komut ise:

**bzip2 –** Sıkıştırma işlemi için Burrows-Wheeler algoritmasını kullanan bir komuttur. Görev bakımından gzip ile tamamen aynı olup işlev bakımından aynı oldukları söylenemez!

Çünkü gzip ile birden fazla dosya sıkıştırılabilirken; bzip2 ile sadece bir tane dosya sıkıştırılabilir. Bu da gzip in kullandığı algoritma olan GNU algoritmasını, Burrows-Wheeler algoritmasına üstün kılar! Bunun yanında, gzip ile birden fazla dosyanın sıkıştırılabilmesi mümkün olduğu için gzip arşivlemede kullanılırken, bzip2 kullanılamaz. Bu yüzden de gzip gibi bir **dosya arşivleyicisi** değildir.

Dosya uzantısı ise: ‘.bz2 (b: Burrows-Wheeler, z: zip)’

**zcat –** Açılımı ‘zip cat’ olan bu aracın kökeni cat komutudur. Yanı .gz, .zip/.z, .bz2 gibi bir uzantıya sahip olan sıkıştırılmış bir dosyanın, o dosyayı açmadan genişletilmesini / dekompres edilmesini (orijinal formuna geri döndürülmesini) ve görüntülenmesini (terminal ekranına yazdırılmasını) sağlar.

Diğer sıkıştırma/genişletme araçlarından farklı olarak, genişletilmiş dosyanın ismini değiştirmez ya da dosyadan .z, .gz., bz2 gibi uzantıları kaldırmaz.

Ayrıca yukarıda da sözü geçtiği üzere diğer araçlardan farklı olarak genişletilmiş çıktıyı, tıpkı cat komutu gibi terminal ekranına yazdırır (görüntüler). Bu yönüyle cat komutuna benzemektedir.

NOT: zcat tam olarak ‘gunzip -c’ ya da ‘gzip -dc’ komutlarına karşılık gelmektedir. Çünkü ‘gzip -d = gunzip’ olduğunu zaten biliyoruz ve yine hatırlarsak gunzip sıkıştırılmış bir dosyayı sadece genişletmeye (eski haline döndürmeye) yarıyorken, zcat dosyayı hem genişletiyor hem de çıktı olan dosyayı bir standart çıktı (terminal çıktısı) haline getiriyor. İşte bu işlevi yerine getirmek için ise gunzip’in -c / --stdout seçeneği mevcuttur. Bu yüzden bu seçenek ile kullanıldığında gunzip’in zcat den bir farkı olmayacaktır!

Sonuç olarak aşağıdaki denklem elde edilir:

gzip -dc = gunzip -c = zcat 🡪 Esasında sadece bu denklem bile zcat’in ne olduğunu bize anlatmada yeterli olacaktır.

ARŞİVLEME

Arşivleme dediğimiz terim esasında, birden fazla aynı amaç için bir araya toplanmış farklı / aynı türde olan dosyalar / dizinler kümesidir.

TANIM 2 (Arşivleme): Birden fazla dosyayı, tek bir paket içerisinde birleştirme işleminin adıdır.

Sıkıştırma ile alakası ise, arşivin amacının uzun vadeli ve düşük maliyetli bir depolama amacı ile oluşturulmalarıdır.

Arşivleme eylemini düşük maliyetli yapan şey, sıkıştırma eylemidir. Bu bağlamda bu iki işlem beraber kullanıldığında depo kullanımı için gayet yararlı bir durum ortaya çıkar. Kısacası biz insanlar, Linux ya da herhangi başka bir işletim sistemi üzerinde yer alan (depolanan) dosyalarımızı uzun süre ve küçük boyutta saklamak amacı ile sıkıştırma & arşivleme eylemlerini (kombine olarak / birlikte) yaparız. Bunu Windows işletim sisteminde sağlayan en ünlü / yaygın kullanımlı araç: hepimizin bildiği, bilmese de duyduğu RAR dır! Linux’ta ise TAR diye kısalttığımız **TAR (Tape Archive)** ve **ZIP’tir**. Türkçe ’de Bantlı / Bantlanmış Arşiv anlamına gelen **TAR,** tıpkı etrafı bir koli bandı ile bantlanmış bir koli gibi düşünülebilir! Zaten arşivlemenin mantığı da bu değil midir?

Tahmin edileceği üzere, tar komutu söz konusu arşivlenecek olan dosyaları sıkıştırmaksızın sadece arşivler.

Ancak hem arşivleyen hem de sıkıştıran başka bir araç var mı derseniz; tar ve gzip komut / araçlarının bir kombinesi olan **TGZ (Tape Archive GNU ZIP)** aracıdır.

Yukarıda bahsedilen iki aracın sırasıyla dosya uzantılar ise: ‘.tar’, ‘.tgz’ şeklindedir.

NOT: Yukarıdaki mantık doğrultusunda; ‘.tgz’, ‘.tar.gz’ (‘.tar’ ve ‘.gz’ kombinesi) uzantısının bir kısaltmasıdır.

NOT: Yukarıdaki iki araç aynı zamanda dosyaları arşivledikleri için (bu işlevlerinden ötürü) onlara bir Dosya Arşivleyicisi de denilir.

Dosya arşivleyicileri, arşivin (dosya ve dizinler kümesinin) boyutunu azaltmak adına kendi arşiv formatlarında Kayıpsız Veri Sıkıştırmasını kullanabilirler. İşte tam da TGZ bu işlevi görür!

**zip –** Linux’taki yukarıda bahsi geçen her iki işlemi (sıkıştırma & arşivleme) de gerçekleştirebilen efsane bir araçtır.

Hatırlarsak tar bir dosya grubunu sadece arşivleyebiliyordu ve sıkıştırma eylemi için gzip gibi bir ekstra sıkıştırma aracına muhtaçtı (TGZ). Ayrıca tar, gzip ile beraber kullanıldığında (TGZ olduğunda); sıkıştırma, tüm arşiv üzerinde uygulanıyordu (her bir dosya için ayrı ayrı değil!). Ancak zip’te sıkıştırma build-in (baştan beri var olan) dir ve arşivde yer alan her dosya için bağımsız olarak gerçekleşir (her bir dosya için ayrı ayrı).

Dosya uzantısı: ‘.zip / .z’

**unzip –** Tahmin edileceği gibi, zip aracının ters işlevini gerçekleştiren araçtır. Yani genişletme işlemini gerçekleştirir. Yine ‘gunzip’ gibi, tersi olan aracın kullandığı uzantıyı kullanır: ‘.zip / .z’