\*Dosyaya yazma işlemi yaparken > kullanılırsa üstüne yazar >> yapılırsa dosya içinde ilave olarak yazar.Cat komutuda nano ve vim gibi kullanılabilir fakat ayrı bir menü açılmaz.cat > dosya.txt

\*head -5 dosya.txt ilk 5 satırı getirir.tail kullanılırsa sondan getirir.

\*cat file1 file2 > dosya.txt 🡪 file1 ve file2 içindeki komut satırlarını doysa.txt ye yazar

\*find /home -name yonetim.txt ile home dizini altında yonetim dosyasının yolunu verir.Dosya adının case sensitive olmasını istemiyorsak -iname yazılmalı

\*find /home -type d -name linux-lessons -> directory(klasör) tipinde linux-lessons adında var olup olmadığını kontrol etme

\*find . -type f -name ‘\*.txt’ 🡪 txt uzantılı tüm dosyaları getirir.Tırnak kullanmak zorunlu.

\*find . -type f -empty 🡪 içi boş klasörleri bulur

\*find /home -size 100M 🡪 size 100mb dan büyük olan dosyaları bulur.

\*find /home -size +50M -size -100M 🡪 50 ile 100 mb arası dosyları bulur.

\*find /home -mtime +10 🡪 10 günden fazladır bulunan dosyları gösterir.(oluşturulma zamanı)

\*grep “kernel” linux.txt 🡪 linux dosyasında kernel kelimelerini aratır.Dosya adı yerine \* koyarsak tüm dosyalara bakar.Case sensitive olmasını istemiyorsak grep -I “kernel” linux.txt yapıyoruz.Bu komut bulunduğu konumdaki file ları kontrol eder.Bulunan dizinde klasör varsa klasör içini kontrol etmez.

\*grep -w “ker” linux.txt yaparsak kernel kelimesini algılamaz sadece kelime olarak “ker” sözcüğü varsa onu bulur.

\*Dosyayı gizli dosya yapmak için mv dosya.txt .dosya.txt ile başına nokta koyarak gizli yapabiliriz.

\*olmayan satırları bulmak için -v kullanılır.

\*grep “^li” dosya.txt 🡪 ^ işareti kelimenin li ile başlaması gerektiğini ifade eder.

\*grep “kernel.$” dosya.txt 🡪 satır sonu kernel. ile biten satırları getirir.

\*grep -A3 “line 5” dosya.txt 🡪 line 5 bulunan satırdan sonra 3 satır daha getir.Önceki satırlar için -B kullanılır.-C ile önce ve sonraki satırları getirir.

\*Linuxun tüm komutlarını /usr/bin/ altında görebiliriz./usr/bin/ls yazarsak yine ls komutunu çalıştırmış oluruz.

\*environment variable,shell variable.ACADEMY=baran 🡪 academy değerine baran valuesunu atamış oluyoruz.Bu shell variable oluyor.Temizlemek istersek bash yazıp enterlıyoruz.Environment variableın değerini okutmak için echo $HOME yada printenv HOME yazarak buluruz. env | grep HOME yazarak bulabiliriz.Shell variablelar terminali kapatıp açtığımızda gider.Env. varlar kalıcıdır.

\*Shell variabları env yazarak görüntüleyemeyiz.set yazarak bulabiliriz.set | grep {shell var. adı} yazarak set çıktısı içinde aratmak istediğimiz shell variablı aratabiliriz.



\*env var oluşturmak için export TEST=name ile yaparız.Env var tüm kullanıcılarda görülür.

\*unset ile shell varın değerini kaldırabiliriz.

\*.sh dosyası oluşturup içine komut satırları yazabiliriz.Fakat yarattığımız .sh dosyasını oluşturduktan sonra execute yetkisi vermemiz gerekir.Bunun için chmod +x test.sh yapmalıyız.

\*Yarattığımız test.sh I execute özelliği ekledikten sonra ./test.sh yazarak çalıştırabiliriz.Fakat ./ kullanmadan bu test.sh filenı path variable olarak eklersek test.sh yazarakta çalıştırabiliriz.Bunun için test.sh ın olduğu klasörün yolunu göstermeliyiz.Bunu yaptıktan sonra test.sh heryerden çağırabiliriz.

export PATH=$PATH:/home/mbaran/Linux/test

\*export ettiğimiz klasör içinde execute edilmeyen file lar varsa bu fileları direkt adıyla çağırırsak permission denied alırız.