

# OykGun 3

Dosya sistemi, diski isletim sis kullanmasi icin ayarlar

. Bulundugumuz dizini gosteren isaret

.. bir ust dizini gosteren isaret

. Bulundugumuz dizini point eder .. bir ust dizini point eder

iki yoldan gidilebilir dizine

1.tam yol(full path)

2.goreceli (relative)

Goreceli yol ls -al ../../var/tmp

Tam yol ls -al ../../var/tmp

Bu da tam yol ls -al /var/tmp

Typo: ufak kullanıcı hataları

Tilda ev dizinini isaret eder /home/gorgen

cd bir program degil kabugun bir fonksiyonudur

cd ile dosya icine giremeyiz sadece dizine girebiliriz

ls e bir yol belirtmezsek bulundugumuz dizini listeler

iki farkli fonk veya programi tek bir komutla birlikte kullanmak istiyorsak mantiksal operatör kullanmsk

lazim

```
ls -al /var/temp cd ../home/gorgen
```

cd ye parametre vermezsek bizi direkt ev dizinine atar  
hangi kullanıcı isek.

cd ye - koyarsak cd den önce nerdeyse oraya atar  
bizi(cd -)

Linux tabanlı işletim sistemlerinde dosya uzantısının hiçbir  
önemi yok

Bunun dosya içindeki veri tipiyle de alakası yok  
file bize dosyanın ne olduğuyla alakalı bilgi verir

whereis çalıştırdığımız programın nerede olduğunu  
gösterir

which de aynı şeyi verir fakat sadece yolunu verir

touch ile aslında zaman damgasını değiştiririz  
dosyanın. Eğer bir dosya yoksa yeni dosya oluşturur

mkdir: make director.

```
mkdir kamp/sistem2/c kamp/sistem2
```

mkdir soldan sağa çalışır

mkdir -p ise eğer gösterdiğimiz yolda var olmayan

dizinler varsa olusturur

ls -R alt directorlari da listeler

whoami (kabuktaki login olunan kullanicinin ismi)

Tum kabuklar ayni zamanda bir programlama dilidir  
hem de bir programlama ortamidir

echo bash ortaminda arguman olarak ne verilmissse  
onun aynisini geri doner

Permission denied:izin verilmedi

nano piconun alternatifi pico linuxda yok  
# yorum satiridir bash de

which bash //hangi komutlari kullanacaksak onun  
dizin yolunu bulup(yorumlayicisini)

nano ile scriptin basina #!dizin yolu yazariz

nano ilkscript

#ls yaz icine

chmod +x ilkscript

./ilkscript (calistirir)

user=\$(whoami)

echo "Merhaba \${whoami}"

eger shebang satirina python yazip icine bash kodlari

yazarsak

bash ilkscrip

Bu çalışır çünkü basta bash dedik

cat: dosyayı görüntüler

Butun programlar derlenmiş halinde kullanılır

pwd\_path=\$(which pwd)

echo \$pwd\_path //pwdnin yolunu ekrana basar

Bir değişken tanımladığımızda o değişken sadece o bash sürecinde çalışır

Logout yaparsak o süreç ölür ve değişkenler boş döner fakat yazdığımız komutlara hala ulaşabiliriz

Bash içinde bash başlatılabilir terminal içinde 2. Bash den çıkış yaparsak birinci bash e döneriz exit

Nano da yazdığımız bir bash script ise bu scripti(programı) yeni bir bash süreci başlatarak çalıştırır. programı bitirdiğimizde o bash sürecide biter ve program içerisindeki değişkenlere ulaşamayız

ls\_path=\$(which ls) bu da ls\_pathin içine ls in yolunu atar

Dat files kullanıcıya özgü ayarları tarihçeleri vs tutar

.bashrc kullaniciya ozgu bash ayar dosyasidir

cat .bashrc

alias definitions bak

Sunucularda alias a dokunmak iyi degil

Sunucularda bash ayar dosyasi degistirilmesi iyi degil

Alias ler ile komutlara takma adlar verilebilir

alias uzun\_listele="ls -l"

uzun\_listele //ls -l calisir

Fakat bulundugumuz bash öldürülürse bu alias de ölür

.bashrc ayar dosyasi oldugu icin daha dikkatli olunmalidir

.bashrc de ki degisikliklerin aktif olabilmesi icin bash oldurup yeniden baslatmaliyiz ya da yeni bash acmaliyiz.

Cunku bash acilirken baslangicta bash ayar dosyasini okuyarak baslar bu bir kere olur

Ya da bu ayar dosyasini tekrar reload edebiliriz bunu da source .bashrc ile yapariz