

BİLGİ İŞLEM

MSGSU FİZİK BÖLÜMÜ

DERS 6

Yrd Doç Dr. Ferhat ÖZOK
MSGSU FİZİK BÖLÜMÜ

If-Else Kalıbı ve Kontrol İşlemleri

Aritmetik karşılaştırmalarda, karakter dizileri (string) ve dosya karşılaştırmalarında aşağıdaki seçenekler kullanılabilir:

Aritmetik		Karakter dizileri		Dosya	
eq	eşit	-z	Boş	-f	Dosya var
gt	büyük	-n	Tanımlı	-s	Dosya boş değil
lt	küçük	=	Eşit	-r	Dosya okunabilir
ge	büyük eşit	!=	Farklı	-w	Dosya yazılabilir
le	küçük eşit			-x	Dosya çalıştırılabilir
				-h	Sembolik bağlantı
				-c	Karakter dosyası
				-b	Blok dosyası

Hemen her programlama dilinde olan if kalıbı bir Linux komutunun çalışmasını kontrol (test) eder. if komutu yerlesik bir komuttur ve her if , bir fi komutuyla bitmelidir. if komutunun ardından gelen Linux komutu çalıştırılır ve komutun çıkış durumu (exit status) gözönüne alı narak ardından gelen then deyimiyle birlikte devamı işletilir. Genellikle komutun iki türlü çıkış durumu olacağından else komutunun ardından gelen komut zinciri, diğer çıkış durumunda çalıştırılır

İki mantıksal sonuç && (ve), || (ya da) ile karşılaştırılabilir

```
if linux komutu
then
komut1
komut2
...
else
komut1
komut2
...
fi
```

Case Döngüsü

Birkaç alternatif arasında seçim yapılması gerekiyorsa case kalıbı kullanılır. Komutun genel kullanımını şu biçimdedir:

```
case aranacak_kelime in  
seçenek1)  
komutlar  
;;  
seçenek2)  
komutlar  
;;  
...  
*)  
komutlar  
;;  
esac
```

Döngüler

Bash kabuk programında aşağıdaki döngü türleri kullanılabilir:

while kontrol ifadesi

do

komutlar

Done

ya da;

for değişken in değerler

do

komutlar

done

1 den 1002e kadar sayan bir shell scripti

```
Terminal
File Edit View Search Terminal Help
GNU nano 2.2.6 File: say.sh
#!/bin/bash
deger=0
while [ $deger -lt 100 ]
do
deger=$((deger+1))
echo $deger
done
[ Read 7 lines ]
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut
```

```
Terminal
File Edit View Search Terminal Help
ozok@ozok-ThinkPad-X200 ~/Desktop/Bilgi_islem_dersi/bashscript $ ./say.sh
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
```

Kullanıcı ya da sistem yöneticisi olarak kullanılan bazı programlarda düzgün deyimlerden yararlanır. Vi editörü içerisinde olduğu gibi grep, sed ve awk programlarında da düzgün deyimler sıklıkla kullanılır.

“abc”, “Merhaba Dünya”, “*ABC*”, “.a?b\$” birer düzgün deyim olarak örneklendirilebilir. Düzgün deyimler içerisinde bazı karakterlerin özel anlamları vardır. Bunlar;

- . Herhangi tek bir karaktere karşılık gelir.
- * Herhangi bir sayıdaki (sifir dahil) herhangi bir karaktere karşılık gelir.
- [..] Küme içindeki karakterlerden herhangi birine karşılık gelir.
- [^...] Küme içindekilerin dışındaki herhangi bir karaktere karşılık gelir.
- ^ Satir başına karşılık gelir.
- \$ Satir sonuna karşılık gelir.
- {n,m} Herhangi bir karakterin bulunma aralığını gösterir.
- {n} Tam olarak n kere olduğunu gösterir.
- {n,} En az n kere olduğunu gösterir.
- {n,m} En az n, en fazla m kere olduğunu gösterir.
- + Bir ya da daha fazla olduğunu gösterir.
- ? Sifir ya da bir kere olduğunu gösterir.
- | Bir önceki ya da sonraki düzgün deyime karşılık gelir.
- () Grup olarak düzgün deyimleri tanımlar.
- \ Özel karakterlerin normal karakter olarak algılanmasını sağlar.

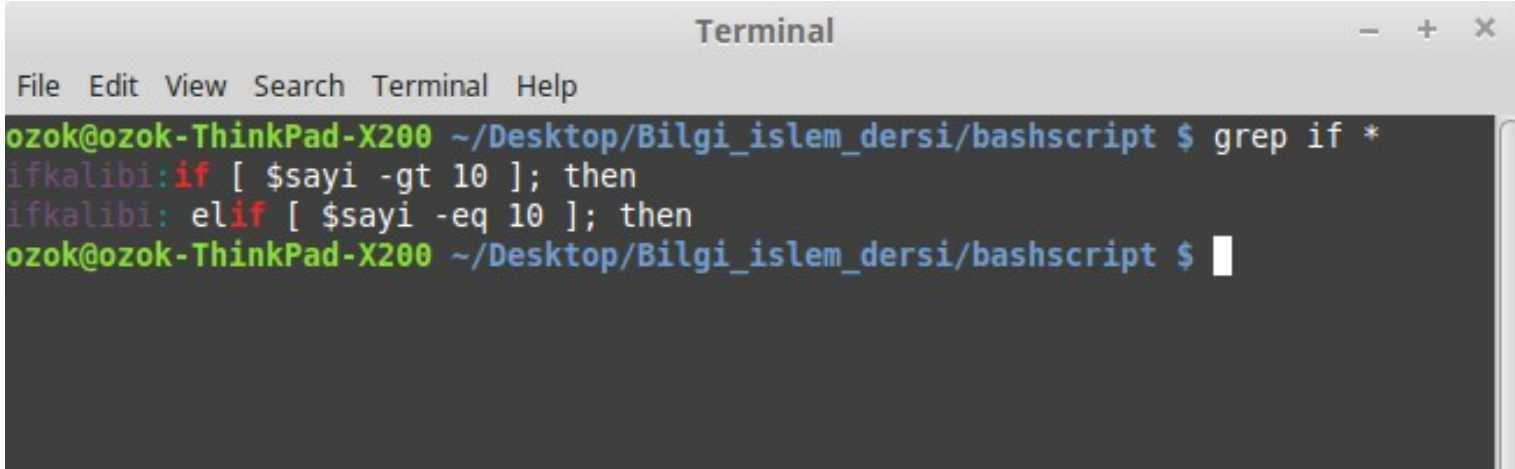
Özel Karakterler

. * [] ^ \$ \ &

grep İLE ÇALIŞMAK

Düzgün deyimlerin en çok kullanıldığı yerlerden birisi dosya içerisinde herhangi bir deyim araştırılmasında kullanılan grep komutudur.

İçerisinde 'if' sözcüğü geçen dosyaları ve 'if' sözcüğünün geçtiği satırı görüntülemek için;



```
Terminal
File Edit View Search Terminal Help
ozok@ozok-ThinkPad-X200 ~/Desktop/Bilgi_islem_dersi/bashscript $ grep if *
ifkalibi:if [ $sayi -gt 10 ]; then
ifkalibi: elif [ $sayi -eq 10 ]; then
ozok@ozok-ThinkPad-X200 ~/Desktop/Bilgi_islem_dersi/bashscript $
```




Satır başında 'do' kelimesi geçen dosyaları bulmak için;

```
Terminal
File Edit View Search Terminal Help
ozok@ozok-ThinkPad-X200 ~/Desktop/Bilgi_islem_dersi/bashscript $ grep '^do' *
dowhile:do
dowhile:done
say.sh:do
say.sh:done
ozok@ozok-ThinkPad-X200 ~/Desktop/Bilgi_islem_dersi/bashscript $
```

Bütün dosyalarda, büyük ve küçük harf ayrımı farketmeksizin, case geçen satırları bulmak için

```
Terminal
File Edit View Search Terminal Help
ozok@ozok-ThinkPad-X200 ~/Desktop/Bilgi_islem_dersi/bashscript $ grep '[Cc][Aa][Ss][Ee]*' *
casedongusu:case $secenek in
islemornek:case $operator in
ozok@ozok-ThinkPad-X200 ~/Desktop/Bilgi_islem_dersi/bashscript $
```

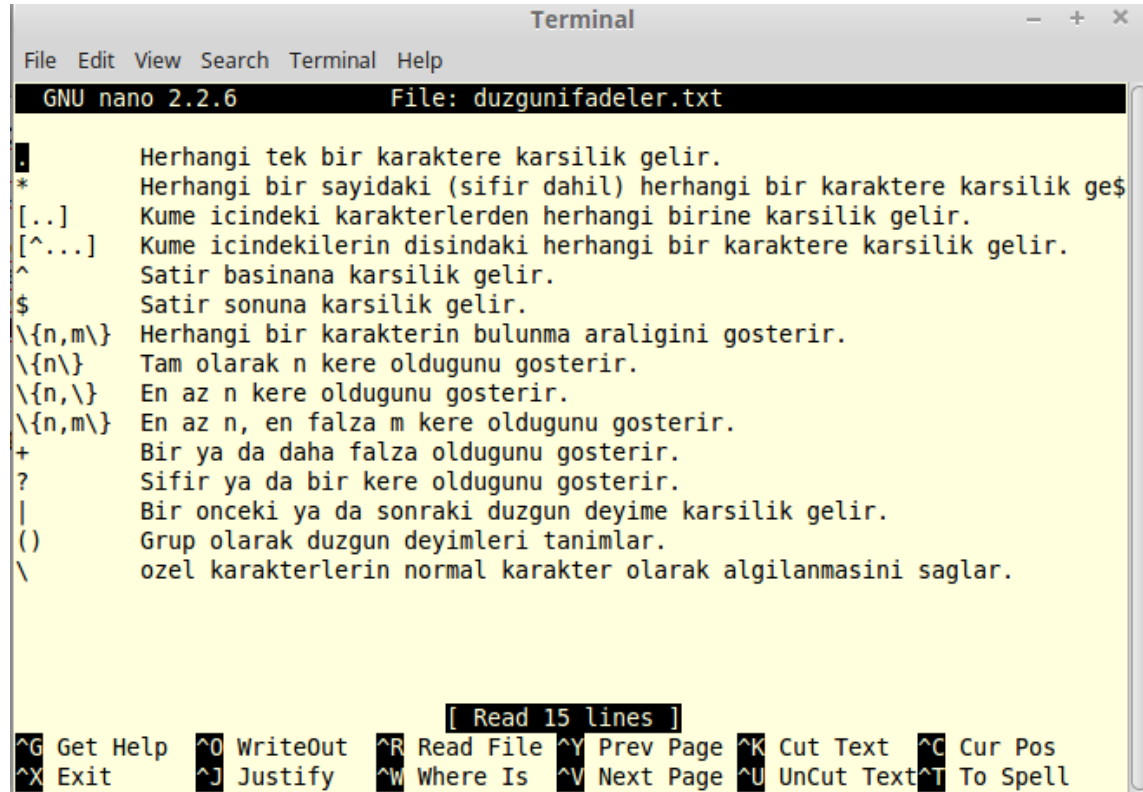
İçinde “.” gecen satirlari aramak icin

```
Terminal
File Edit View Search Terminal Help
ozok@ozok-ThinkPad-X200 ~/Desktop/Bilgi_islem_dersi/bashscript $ grep '\.' *
fortest:#for i in {1..5}
fortest:echo "Bash version ${BASH_VERSION}..."
fortest:for i in {0..10..2}
islemornek:echo "1. sayi : $s1"
islemornek:echo "2. sayi : $s2"
ozok@ozok-ThinkPad-X200 ~/Desktop/Bilgi_islem_dersi/bashscript $ █
```

sed İLE ÇALIŞMAK

Sed programı 'Stream Editor' anlamına gelip metin belgeleri üzerinde komutlar ile değişiklikler yapmayı sağlar. Örneğin bir dosyadaki 'van' kelimelerini 'one' ile değiştirmek için 's/van/one/g' komutu kullanılabilir. Komuttaki s değiştir (substitute), g genel (global) anlamına gelmektedir. Sed, belirtilen komutları girilen dosya üzerinde çalıştırır ve çıktıları ekrana ya da belirtilen dosyaya yönlendirir. Sed içerisinde de düzgün deyimler kullanılabilir. Örneğin satır başındaki boşlukların silinmesi için 's/^ */' komutu girilir.

Aynı anda birden çok değiştirme yapabilmek için ise her argüman için -e kullanılır.
sed -e 's/computer/bilgisayar/g' -e 's/server/sunucu/g' dosya



```
Terminal
File Edit View Search Terminal Help
GNU nano 2.2.6 File: duzgunifadeler.txt
| Herhangi tek bir karaktere karşılık gelir.
* Herhangi bir sayıdaki (sifir dahil) herhangi bir karaktere karşılık ge$
[...] Kume içindeki karakterlerden herhangi birine karşılık gelir.
[^...] Kume içindekilerin dışındaki herhangi bir karaktere karşılık gelir.
^ Satır başına karşılık gelir.
$ Satır sonuna karşılık gelir.
\{n,m\} Herhangi bir karakterin bulunma aralığını gösterir.
\{n\} Tam olarak n kere olduğunu gösterir.
\{n,\} En az n kere olduğunu gösterir.
\{n,m\} En az n, en fazla m kere olduğunu gösterir.
+ Bir ya da daha fazla olduğunu gösterir.
? Sifir ya da bir kere olduğunu gösterir.
| Bir önceki ya da sonraki düzgün deyimle karşılık gelir.
() Grup olarak düzgün deyimleri tanımlar.
\ özel karakterlerin normal karakter olarak algılanmasını sağlar.

[ Read 15 lines ]
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

```
ozok@ozok-ThinkPad-X200 ~/Desktop/Bilgi_islem_dersi/bashscript $ sed 's/fälza/fazla/g' duzgunifadeler.txt
.      Herhangi tek bir karaktere karsilik gelir.
*      Herhangi bir sayidaki (sifir dahil) herhangi bir karaktere karsilik gelir.
[..]   Kume icindeki karakterlerden herhangi birine karsilik gelir.
[^...] Kume icindekilerin disindaki herhangi bir karaktere karsilik gelir.
^      Satir basinana karsilik gelir.
$      Satir sonuna karsilik gelir.
\{n,m\} Herhangi bir karakterin bulunma araligini gosterir.
\{n\}   Tam olarak n kere oldugunu gosterir.
\{n,\}  En az n kere oldugunu gosterir.
\{n,m\} En az n, en fazla m kere oldugunu gosterir.
+      Bir ya da daha fazla oldugunu gosterir.
?      Sifir ya da bir kere oldugunu gosterir.
|      Bir onceki ya da sonraki duzgun deyime karsilik gelir.
()     Grup olarak duzgun deyimleri tanimlar.
\      ozel karakterlerin normal karakter olarak algilanmasini saglar.
```