



This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-nc-sa/4.0/).

BilkentGNU Linux Eğitimi 3. Hafta

- Grep, Regular Expressions, Find



grep

- Herhangi bir dosyanın içinde arama yapmanızı sağlar, açılımı **"Global Regular Expression Parser"**
- Şu anda Regular Expressions konusunu işlemedik fakat grep'e giriş yapıp sonrasında bu konuya değineceğiz.

```
$ grep admin cut_sample.txt
```

- Önceki dersten öğrendiğimiz wildcard'ları hatırlayalım.
 - "?": herhangi bir karakter.
 - "*": herhangi bir karakter(ler).

```
$ grep admin *.txt
```

- grep öğrendiğimiz pipe operasyonu ile sık sık kullanılan bir komut.

```
$ ls -l | grep hello
```

Regular Expressions

- Uzun bir text içerisinde belli bir yapıyı aramak için kullanılan sembol ve karakterler topluluğu.
- Çoğunlukla regex **olarak** kısalatılır
- Arama motorları, metin editörleri veya text inputu bulunan bir çok yazılımda regex motorları bulunur.
- Hemen hemen her programa dilinde en az bir tane **regex** kütüphanesi vardır.

Regular Expressions

- "^" Satır başını gösterir.
- "\$" Satır sonunu gösterir.

```
$ grep "^GNU" filename
```

```
$ grep "GNU$" filename
```

Regular Expressions

- Örnek karakter kümeleri: [A-Z], [a-z], [A-F], [0-9]
- "[]" içinde bulunan karakter kümesinden herhangi bir karakter.
- "[^]" içinde bulunan karakter kümesinde bulunmayan herhangi bir karakter.

```
$ grep "c[au]t" filename
```

```
$ grep "[^windows]" filename
```

```
$ grep "[1-9][0-9]" filename
```

Regular Expressions

- "*" wildcard kullanımı dışında kendinden önceki ifadeyi tekrar etmek için kullanılır.

```
$ grep "[A-Z]*" filename
```

```
$ grep "^ [A-Za-z].*\.$" filename
```

Regular Expressions

- "{", "*" karakterinin yaptığı işi sayı ile sınırlamamızı sağlar.
- {a}: kendinden önceki ifade **a** kere tekrar etmeli.
- {a,b}: kendinden önceki ifade ne az **a** en çok **b** kere tekrar etmeli(a ve b dahil).
- {a, }: kendinden önceki ifade en az **a** kere tekrar etmeli.
- { ,b}: kendinden önceki ifade en çok **b** kere tekrar etmeli.
- "{" ve "}" yazarken "\"(escape character) kullanılmalıdır.

```
$ grep "[A-Z]\{5\}" filename
```

- Daha fazlası için "**grep -E**" (Extended grep)

Regex Alıştırmaları

- Kullanıcı adı: en az 4 en çok 16 karakter, "-" ve "_" içerebilir)
- Parola: en az 8 en çok 16 karakter, herhangi bir noktalama işareti içerebilir)
- Telefon numarası
- URL
- Email adresi

find

- **man find:** walk a file hierarchy.
- Belirtilen bir dizin de recursive arama yapmanızı sağlar.
- Genel kullanım formatı:

```
$ find directory_name options other
```

find

- "-name"
- Belirli bir dosya ismini listelemek için kullanılır.

```
$ find . -name test.txt
```

```
$ find . -iname test.txt
```

find

- "-type"
- Aranan dosyanın tipini belirtmek için kullanılır.

```
$ find / -type d -name week
```

```
$ find . -type f -name test.txt
```

```
$ find . -type f -name "*.txt"
```

find

- "-size"
- Aranan dosyanın boyutunu belirtmek için kullanılır.

```
$ find / -size 10M
```

```
$ find / -size +10M
```

```
$ find / -size +10M -size -30M
```

find

- "-exec"
- Arama sonuçlarında bulunan dosyalara herhangi bir işlem uygulamak için kullanılır

```
$ find / -size 10M -exec rm -rf {} \;
```

3. Haftanın Sonu

- Tebrikler, BilkentGNU Linux Eğitiminin üçüncü haftası sona erdi.
- Bu derste öğrendiğimiz komut ve konseptleri tekrar ederek kafanızda yer etmesini sağlayabilirsiniz.

