



Day 9 Notes

- `cat -n count.txt` (n flag'i kullanıldığında her satırın numarasını da gösteriyor)
- `cat -n count.txt >> yeni.txt` → bu satırlı olanı yeni bir dosya yaratıp ona yazar
- `tac count.txt | tee temp.txt` → `tac count.txt` çıktısını `temp.txt` diye bir dosya yaratıp ona yazdırıyor
- `cat > tennis.txt` → dosyayı editleyebiliyorsun, çıkmak için `ctrl+d`
- `grep Williams` → arama
- `cat tennis.txt | grep Williams`
- `ls -l | cut -d' ' -f3` → listelemede boşluğu referans olarak üçüncü sütunu getir
- `cat << EOF > yonetim.txt` (EOF, end of file, dosyayı kapatıyor)
Yonetim:Turkiye'nin resmi egitim partneri.
EOF
- `cat yonetim.txt | tr oet QAZ` → Dosyanın çıktısına oet harfleri yerine QAZ yaz
- `\n` → alt satıra geçişi sağlar
- `cat count.txt | tr '\n' ' '` → alt satıra geçmek yerine boşlukla değiştir
- `cat count.txt | tr [a-z] [A-Z]` → küçük yazıyı büyük yapar
- `wc count.txt` → word count, alt satıra geçmek de sayılıyor
- `sort` komutu sıralıyor. -r flag'i?
- `sort trainees.txt | uniq` → çiftleyenleri siliyor
-

```

oguzhan_91@DESKTOP-C9CE1DQ:~/filters$ cat countries.csv | cut -d"," -f2
Capital
Washington
Paris
Ottawa
Berlin
oguzhan_91@DESKTOP-C9CE1DQ:~/filters$ cat countries.csv | cut -d"," -f3
Continent
North America
Europe
North America
Europe
oguzhan_91@DESKTOP-C9CE1DQ:~/filters$ cat countries.csv | cut -d"," -f3 | grep -v "Continent"
North America
Europe
North America
Europe
oguzhan_91@DESKTOP-C9CE1DQ:~/filters$ cat countries.csv | cut -d"," -f3 | grep -v "Continent" | sort
Europe
Europe
Europe
North America
North America
oguzhan_91@DESKTOP-C9CE1DQ:~/filters$ cat countries.csv | cut -d"," -f3 | grep -v "Continent" | sort | uniq
Europe
North America
oguzhan_91@DESKTOP-C9CE1DQ:~/filters$ cat countries.csv | cut -d"," -f3 | grep -v "Continent" | sort | uniq | tee continents.txt
Europe
North America
oguzhan_91@DESKTOP-C9CE1DQ:~/filters$ cat continents.txt
Europe
North America
oguzhan_91@DESKTOP-C9CE1DQ:~/filters$

```

Control Operator	Kullanımı
; <u>semicolon</u>	Birbirinden bağımsız komutları tek satırda kullanmamıza yarıyor.
& <u>ampersand</u>	Bu işaret, bir komutun çalıştırıldığında, komutun hemen ardından yazılan komutun beklemeden çalıştırılmasını sağlar.
\$? <u>dollar question mark</u>	Bir komutu çalıştırdıktan sonra "\$?" ifadesini kullanarak, komutun başarılı bir şekilde tamamlanıp tamamlanmadığını kontrol edebilirsiniz.
&& <u>double ampersand</u>	İki komutu ardışık olarak çalıştırmak ve bir sonraki komutun yalnızca önceki komut başarılı bir şekilde tamamlandığında çalışmasını sağlamak için kullanılır.
 <u>double vertical bar</u>	İki komutu ardışık olarak çalıştırırken, bir sonraki komutun yalnızca önceki komut hatalı bir şekilde tamamlandığında çalışmasını sağlar.
Combining && and 	"&&" ve " " işaretlerini birleştirerek, koşullu ve alternatif işlemler oluşturabilir ve karmaşık komut zincirlemeleri gerçekleştirebilirsiniz.
# <u>pound sign</u>	Yorum satırı veya komut dosyasında açıklama yapmak için kullanılan bir semboldür.

; sırayla yaptırıyor, & aynı anda

```

oguzhan_91@DESKTOP-C9CE1DQ:~/shell-scripting$ nano basic.sh
oguzhan_91@DESKTOP-C9CE1DQ:~/shell-scripting$ ls -l
total 4
-rw-rw-r-- 1 oguzhan_91 oguzhan_91 31 Jul 13 14:30 basic.sh
oguzhan_91@DESKTOP-C9CE1DQ:~/shell-scripting$ chmod +x basic.s
oguzhan_91@DESKTOP-C9CE1DQ:~/shell-scripting$ ls -l
total 4
-rwxrwxr-x 1 oguzhan_91 oguzhan_91 31 Jul 13 14:30 basic.sh
oguzhan_91@DESKTOP-C9CE1DQ:~/shell-scripting$ ./basic.sh
Hello World
oguzhan_91@DESKTOP-C9CE1DQ:~/shell-scripting$ cat basic.sh
#!/bin/bash
echo "Hello World"
oguzhan_91@DESKTOP-C9CE1DQ:~/shell-scripting$ nano |

```

- `#!/bin/bash` hangi shell olduğunu belirtiyor. shell dosyasını çalıştırmak için `./dosya` adı.
- absolute mode: numaralarla permisson vermek 757 gibi
- `grep` is a command that helps to search content and display them according to the user-specified regular expression while the `find` command helps to search and locate files according to the given criteria.
- sh dosyalarını `./` koymadan çalıştırmak için path olarak eklememiz gerekiyor
 - `export PATH=$PATH:/home/mserbas/shell-scripting`
 - `~/shell-scripting$`

```
oguzhan_91@DESKTOP-C9CE1DQ:~/shell-scripting$ nano basic.sh
oguzhan_91@DESKTOP-C9CE1DQ:~/shell-scripting$ ./basic.sh
Hello World
Thu Jul 13 14:37:25 +03 2023
/home/oguzhan_91/shell-scripting
oguzhan_91@DESKTOP-C9CE1DQ:~/shell-scripting$ export PATH=$PATH:/home/oguzhan_91/shell-scripting
oguzhan_91@DESKTOP-C9CE1DQ:~/shell-scripting$ basic.sh
Hello World
Thu Jul 13 14:40:04 +03 2023
/home/oguzhan_91/shell-scripting
basic.sh
oguzhan_91@DESKTOP-C9CE1DQ:~/shell-scripting$ cd ..
oguzhan_91@DESKTOP-C9CE1DQ:~$ basic.sh
Hello World
Thu Jul 13 14:40:10 +03 2023
/home/oguzhan_91
shell-scripting
oguzhan_91@DESKTOP-C9CE1DQ:~$ cd ..
oguzhan_91@DESKTOP-C9CE1DQ:/home$ basic.sh
Hello World
Thu Jul 13 14:40:15 +03 2023
/home
oguzhan_91 user1
oguzhan_91@DESKTOP-C9CE1DQ:/home$ |
```

```
oguzhan_91@DESKTOP-C9CE1DQ:~/shell-scripting$ nano variable.sh
oguzhan_91@DESKTOP-C9CE1DQ:~/shell-scripting$ cat variable.sh
#!/bin/bash
echo "Enter your name: "
read NAME
echo "Welcome $NAME"
oguzhan_91@DESKTOP-C9CE1DQ:~/shell-scripting$ variable.sh
Enter your name:
Oguzhan
Welcome Oguzhan
oguzhan_91@DESKTOP-C9CE1DQ:~/shell-scripting$ |
```

```
oguzhan_91@DESKTOP-C9CE1DQ:~/shell-scripting$ read -p "Enter your name: " NAME
Enter your name: Oguzhan
oguzhan_91@DESKTOP-C9CE1DQ:~/shell-scripting$ variable.sh
Enter your name:
Oguzhan
Welcome Oguzhan
oguzhan_91@DESKTOP-C9CE1DQ:~/shell-scripting$ read -p "Enter your name: " NAME
Enter your name: Oguzhan
oguzhan_91@DESKTOP-C9CE1DQ:~/shell-scripting$ echo "Welcome $NAME"
Welcome Oguzhan
oguzhan_91@DESKTOP-C9CE1DQ:~/shell-scripting$ read -s -p "Enter your password: " PASSWORD
Enter your password: oguzhan_91@DESKTOP-C9CE1DQ:~/shell-scripting$ |
```

```
GNU nano 6.2
#!/bin/bash
echo "File Name is $0"
echo "First Parameter is $1"
echo "Second Parameter is $2"
echo "Third Parameter is $3"
echo "All the Parameters are $@"
echo "Total Number of Parameters : $#"
```

`echo "$RANDOM is a random number"`

`echo "The current line number is $LINENO"`

- `expr 5 + 3` hesaplama yapar
 - `expr 3 + 5`
`expr 6 - 2`
`expr 7 * 3`
`expr 9 / 3`
`expr 7 % 2`
 - `#!/bin/bash`
`read -p "Input first number: " first_number`
`read -p "Input second number: " second_number`
`echo "SUM=" expr $first_number + $second_number`
`echo "SUB=" expr $first_number - $second_number`
`echo "MUL=" expr $first_number * $second_number`
`echo "DIV=" expr $first_number / $second_number`