

## ► stdin, stdout, stderr

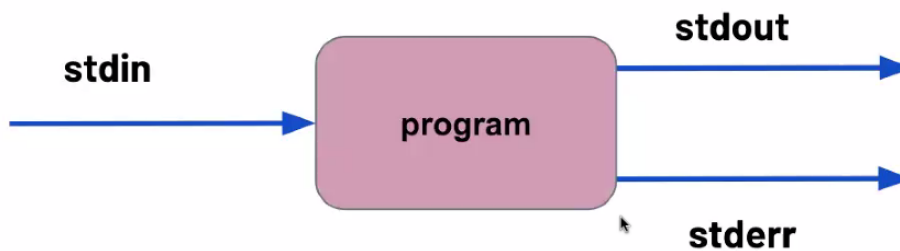
## ► Filters

## ► Commands:

- cat, tee, grep, cut, tr, wc, sort, uniq, comm

## ► Control Operators

Stdin, stdout ve stderr:



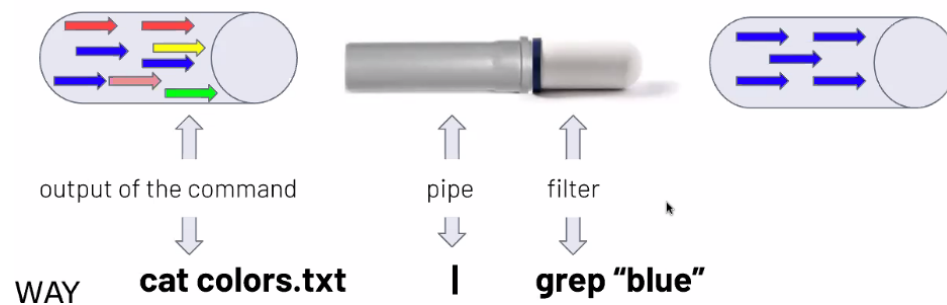
Stdin, Input kısmı

Stdout, 1 nolu kanal output, kod ya da program çalıştığındaki output

Stderr, 2 nolu kanal error, kod ya da program çalışmazsa error

Linux te komutları bir araya getiriyoruz ve koyduğumuz sıraya göre veriyi tek tek işleyip arzu ettiğimiz çıktıyı bize veriyorlar.

Pipe önceki komuttan aldığı outputu sonraki komuta input olarak aktarır.



Pipe kullanmanın sayı olarak sınırı yok.

cat

When between two pipes, the cat command does nothing (except putting stdin on stdout). Displays the text of the file line by line.

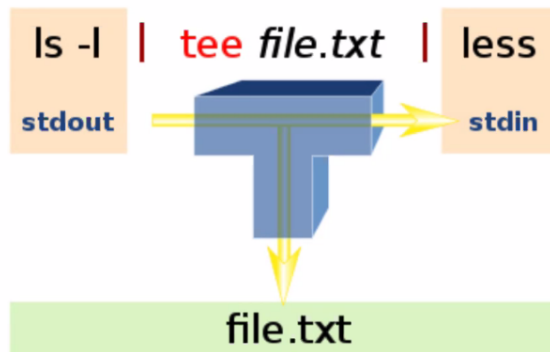
```
ubuntu@clarusway:~$ cat days.txt
sunday
monday
tuesday
wednesday
thursday
friday
saturday
ubuntu@clarusway:~$ cat days.txt | cat | cat | cat | cat
sunday
monday
tuesday
wednesday
thursday
friday
saturday
ubuntu@clarusway:~$
```

Input olarak bir dosyayı alıp içeriğini yansıtır cat komutu.

tee komutu:

tee

tee is almost the same as cat, except that it has two identical outputs.



cat komutu kendisine verilen dosyayı yazdırıyor. tee komutu kendisine gelen veriyi (stdin) verilen dosyanın içine yazar ve veriyi, kendisinden sonra gelen pipe işaretine devreder. Fotokopiciyi düşünün, önüne önemli bir not geldi, tee komutuyla veriyi sakladı ve fotokopiyi başka bir öğrenciye verdi.

Örnekte;

```
ubuntu@ip-172-31-92-42:~$ ls -l
total 20
-rw-rw-r-- 1 ubuntu ubuntu 10141 Dec 14 17:57 README.md
-rw-rw-r-- 1 ubuntu ubuntu 8177 Dec 14 17:58 sed-awk.md
ubuntu@ip-172-31-92-42:~$ ls -l | tee tee.txt | grep "-rw"
Usage: grep [OPTION]... PATTERNS [FILE]...
Try 'grep --help' for more information.
ubuntu@ip-172-31-92-42:~$ ls -l | tee tee.txt | grep -i sed
-rw-rw-r-- 1 ubuntu ubuntu 8177 Dec 14 17:58 sed-awk.md
ubuntu@ip-172-31-92-42:~$
```

ls-l ile ekrana yansıyan veriyi tee komutuyla tee.txt içine yazdırdık ve grep komutuyla da sed geçen satırı getirdik.

```
tee.txt x
tee.txt
1 total 20
2 -rw-rw-r-- 1 ubuntu ubuntu 10141 Dec 14 17:57 README.md
3 -rw-rw-r-- 1 ubuntu ubuntu 8177 Dec 14 17:58 sed-awk.md
4 -rw-rw-r-- 1 ubuntu ubuntu 0 Dec 14 18:24 tee.txt
5
```

grep komutu:

grep

The most common use of grep is to filter lines of text containing (or not containing) a certain string.

```
ubuntu@clarusway:~$ cat tennis.txt
Amelie Mauresmo, Fra
Justine Henin, BEL
Serena Williams, USA
Venus Williams, USA
ubuntu@clarusway:~$ cat tennis.txt | grep Williams
Serena Williams, USA
Venus Williams, USA
ubuntu@clarusway:~$
```

grep istediğimiz kelimeyi, stringi filtreler.

cut komutu:

cut

The cut filter can select columns from files, depending on a delimiter or a count of bytes

`cut -d(delimiter) -f(columnNumber) <fileName>`

```
user@clarusway-linux:~$ ls *.* -l
-rw-r--r-- 1 user user 16 Mar 2 21:56 classes.html
-rw-r--r-- 1 user user 8980 Mar 2 21:53 examples.desktop
-rw-r--r-- 1 user user 24 Mar 2 23:22 html.txt
-rw-r--r-- 1 user user 17 Mar 2 22:42 lesson.txt
-rw-r--r-- 1 user user 13 Mar 2 23:22 linux.txt
-rw-r--r-- 1 user user 0 Mar 4 21:42 xtml.txt
user@clarusway-linux:~$ ls *.* -l | cut -d' ' -f3
user
user
user
user
user
user
user@clarusway-linux:~$
```

cut komutu bir sütunu filtreler:

-d delimiteri yani ayraç olarak kullanılan simgeyi simgeler. bu örnekte boşluk. : olabilir . olabilir , olabilir.

-f sütun numarasını simgeler bu iki değeri vermek gerekir.

passwd dosyasının içeriğinden 1. sütunu listeleyelim:

```
ubuntu@ip-172-31-92-42:~$ cat /etc/passwd | cut -d':' -f1
root
daemon
bin
sys
sync
games
man
lp
mail
news
uucp
proxy
www-data
```

bir pipe daha ekleyerek grep komutu kullanalım:

```
ubuntu@ip-172-31-92-42:~$ cat /etc/passwd | cut -d':' -f1 | grep sshd
sshd
ubuntu@ip-172-31-92-42:~$
```

tr komutu:

tr

The command 'tr' stands for 'translate'. It is used to translate, like from lowercase to uppercase and vice versa or new lines into spaces.

```
ubuntu@clarusway:~$ cat clarusway.txt
Way to Reinvent Yourself
ubuntu@clarusway:~$ cat clarusway.txt | tr 'aer' 'iou'
Wiy to Roinvont Youusolf
ubuntu@clarusway:~$ cat count.txt
one
two
three
four
five
ubuntu@clarusway:~$ cat count.txt | tr '\n' ' '
one two three four five ubuntu@clarusway:~$
```

translate komutu. çalıştığımız bir kelime ya da cümle ya da uzun bir text içerisinde bir harfin karşılığı yerine başka bir harfi getirmek gibi bir işlevi vardır.

üstteki komutta a yerine i e yerine o r yerine u koy diyoruz.

alttaki komutta new line yerine space bırak diyoruz. bu değişiklikleri ara bellekte yapıyor, yani bize yansıtıyor. Dosya içeriğini değiştirmiyor.

tr komutu karakter silebilir ya da büyük harfleri küçük harfe küçük harfleri büyük harfe çevirebilir:

```

ubuntu@ip-172-31-92-42:~/filters$ cat clarusway.txt | tr -d aeiou
Clrswy:Rd t rnvnt yrslf.
ubuntu@ip-172-31-92-42:~/filters$ cat clarusway.txt | tr [a-z] [A-Z]
CLARUSWAY:ROAD TO REINVENT YOURSELF.
ubuntu@ip-172-31-92-42:~/filters$

```

mesela \_ ile yazılan bir içeriği sistem \_ kabul etmediğinde - ile değiştirebiliyoruz.

wc komutu:



Counting words, lines and characters is easy with wc.

- wc <fileName> (Counts words, lines and characters)
- wc -l <fileName> (Counts only lines)
- wc -w <fileName> (Counts only words)
- wc -c <fileName> (Counts only characters)

```

ubuntu@clarusway:~$ cat count.txt
one
two
three
four
five
ubuntu@clarusway:~$ wc count.txt
 5  5 24 count.txt
ubuntu@clarusway:~$ wc -l count.txt
5 count.txt
ubuntu@clarusway:~$ wc -w count.txt
5 count.txt
ubuntu@clarusway:~$ wc -c count.txt
24 count.txt
ubuntu@clarusway:~$

```

bir dosya içeriğindeki kelimeleri, satırları, karakterleri saydırmak için kullanılır.  
üstteki komutta 24 karakterin nedeni:

hepsinin sonunda aslında gizli \0 var

wc count şeklinde yazıldığında verdiği bilgide ilk sırada satır, sonra kelime ve karakter sayısını yansıtır.

sort komutu:

karışık yazılmış bir veri setini alfabetik sıraya dizebilir. Düz veya ters dizer:

sort

The sort filter will default to an alphabetical sort.

sort -r	the flag returns the results in reverse order
sort -f	the flag does case insensitive sorting

```
ubuntu@clarusway:~$ cat marks.txt
John-10
James-9
Aaron-8
Oliver-7
Walter-6
ubuntu@clarusway:~$ sort marks.txt
Aaron-8
James-9
John-10
Oliver-7
Walter-6
ubuntu@clarusway:~$
```

-r ile ters alfabetik sırayla dizer.

-f option u büyük-küçük harf i dikkate almadan dizmesini sağlar.

uniq komutu:

uniq

With the help of uniq command you can form a **sorted list** in which every word will occur only once.

```
ubuntu@clarusway:~$ cat marks.txt
John
James
Aaron
Oliver
Walter
Aaron
John
James
John
John

ubuntu@clarusway:~$ sort marks.txt | uniq

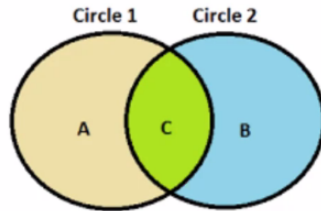
Aaron
James
John
Oliver
Walter
ubuntu@clarusway:~$
```

bir dosya içeriğindeki her kelimenin bir kere gösterilmesini istediğimizde kullanırız. Önce alfabetik sıraya dizer sonra tekrar edenleri siler.

comm komutu:

comm

The 'comm' command compares two files or streams. By default, 'comm' will always display three columns. First column indicates non-matching items of first file, second column indicates non-matching items of second file, and third column indicates matching items of both the files. Both the files has to be in sorted order for 'comm' command to be executed.



```
ubuntu@clarusway:~$ cat file1
Aaron
James
John
Oliver
Walter
ubuntu@clarusway:~$ cat file2
Guile
James
John
Raymond
ubuntu@clarusway:~$ comm file1 file2
Aaron
      Guile
      James
      John
Oliver
      Raymond
Walter
ubuntu@clarusway:~$
```

CLARUSWAY

elimizdeki iki dosyayı karşılaştırdı. default comm üç tane sütun gösterir. Birinci sütun sadece birinci dosyada öğeleri, ikinci sütun sadece ikinci dosyadaki öğeleri, üçüncü sütun ise kesişimde olan öğeleri listeler.

Exercise:

1. Create a file named countries.csv with the following content

```
Country,Capital,Continent
USA,Washington,North America
France,Paris,Europe
Canada,Ottawa,North America
Germany,Berlin,Europe
```

2. a. Cut only "Continent" column  
b. Remove header  
c. Sort the output  
d. List distinct values  
e. Save final output to "continents.txt" file
3. Display content of continents.txt file

```
ubuntu@ip-172-31-89-213:~$ cat countries.csv | cut -d',' -f3 | tail -4 | sort | uniq
Europe
North America
ubuntu@ip-172-31-89-213:~$
```

Control Operatörler:

We put more than one command on the command line using control operators.

Control Operator	Usage
; semicolon	More than one command can be used in a single line.
& ampersand	Command ends with & and doesn't wait for the command to finish.
\$? dollar question mark	Used to store exit code of the previous command.
&& double ampersand	Used as logical AND.
double vertical bar	Used as logical OR.
Combining && and	Used to write if then else structure in the command line.
# pound sign	Anything was written after # will be ignored.

; birbirinden bağımsız birden fazla komutu tek satıra yazmamızı sağlayan bir komut:

```
ubuntu@ip-172-31-92-42:~$ ls -l; sudo apt update -y ; echo "Updated successfully" ; sleep 3
total 28
-rw-rw-r-- 1 ubuntu ubuntu 10141 Dec 14 17:57 README.md
-rw-rw-r-- 1 ubuntu ubuntu 134 Dec 14 19:23 countries.csv
-rw-rw-r-- 1 ubuntu ubuntu 8177 Dec 14 17:58 sed-awk.md
-rw-rw-r-- 1 ubuntu ubuntu 176 Dec 14 18:24 tee.txt
-rw-rw-r-- 1 ubuntu ubuntu 0 Dec 14 18:32 text.txt
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Fetched 110 kB in 0s (300 kB/s)
```

& terminalin bizden alındığı durumlarda & işaretle o process i arka tarfa gönderiyoruz ve terminali kullanmaya devam edebiliriz. komuta bir numara verir:

```
ubuntu@ip-172-31-89-213:~$ sleep 20 &
[1] 1815
```

\$? environment variable da bir önceki yazmış olduğumuz komutun sonuçlarını veriyor. Yani ya başarılı mı değil mi onu bildirir.

0 rakamı verdiğinde bir önce çalıştırdığımız komutun başarılı şekilde çalıştığını söyler.

```
ubuntu@ip-172-31-92-42:~$ cat countries.csv
[1]+  Done                  sleep 10
Country, Capital, Continent
USA, Washington, North-America
France, Paris, Europe
Canada, Ottawa, North-America
Germany, Berlin, Europeubuntu@ip-172-31-92-42:~$ echo $?
0
ubuntu@ip-172-31-92-42:~$
```



olmayan bir komut yazalım:

```
ubuntu@ip-172-31-92-42:~$ dls --l
Command 'dls' not found, did you mean:
  command 'rls' from snap rustup (1.24.3)
  command 'ols' from deb speech-tools (1:2.5.0-12)
  command 'dcs' from deb drbl (4.5.16-1)
  command 'dll' from deb brickos (0.9.0.dfsg-12.2)
  command 'als' from deb atool (0.39.0-11)
  command 'hls' from deb hfsutils (3.2.6-15build2)
  command 'ils' from deb sleuthkit (4.11.1+dfsg-1)
  command 'dlv' from deb delve (1.8.1-1)
  command 'ls' from deb coreutils (8.32-4.1ubuntu1)
  command 'dms' from deb anacrolix-dms (1.1.0-2)
  command 'jls' from deb sleuthkit (4.11.1+dfsg-1)
  command 'fls' from deb sleuthkit (4.11.1+dfsg-1)
  command 'dlg' from deb pccts (1.33MR33-6.1)
See 'snap info <snapname>' for additional versions.
ubuntu@ip-172-31-92-42:~$ echo $?
127
ubuntu@ip-172-31-92-42:~$
```

komutlar her zaman bir rakam tutarlar. 0 dan farklı bir değer aldığında bu bir hata ile karşılaşmıştır. dls komutu olmadığı için \$? bize 0 dan farklı bir rakam verdi.

```
Exit Code | Meaning |
|:-----|:-----|
|1| Catchall for general errors
|2| Misuse of shell builtins
|126| Command invoked cannot execute
|127| Command not found
|128| Invalid argument to exit
|128+n| Fatal error signal "n"
|255| Exit status out of range (exit takes only integer args in the range 0 - 255)
```

&& and demektir. önceki komutun sonucu 0 sa (yani komut çalışıyorsa) sonra gelen komutu da çalıştırır. İlk komut çalışmazsa ikincisi de çalışmaz:

```
ubuntu@ip-172-31-92-42:~$ cat kountries.csv && sudo apt update -y
cat: kountries.csv: No such file or directory
ubuntu@ip-172-31-92-42:~$
```

||or demektir. ilk komut çalışmazsa ikinci komut çalışır:

```
ubuntu@ip-172-31-92-42:~$ cat kountries.csv || sudo apt update -y
cat: kountries.csv: No such file or directory
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
1 package can be upgraded. Run 'apt list --upgradable' to see it.
ubuntu@ip-172-31-92-42:~$
```

&& ve || if else yapılarını kullanmak için bir arada kullanılır.

# comment eklemek için kullanılır. # dan gelen sonra ifadeler komut olarak görmezden gelinir.

```
ubuntu@clarusway:~$ mkdir test # We create a directory
ubuntu@clarusway:~$ cd test # We enter the directory
ubuntu@clarusway:~/test$ ls # is it empty ?
ubuntu@clarusway:~/test$
```

\ karakteri süper güçlü bir karakterin süper güçlerini elinden alır.

```
ubuntu@clarusway:~$ echo this is \* symbol.
this is * symbol.
ubuntu@clarusway:~$ echo this \ \ \ \is \ \ \ \clarusway.
this is clarusway.
ubuntu@clarusway:~$ echo escaping \\ \# \& \\" \ '
escaping \ # & " '
ubuntu@clarusway:~$
```

\ bir diğer fonksiyonu da alt satıra geç komutu çalıştırma demek.

```
ubuntu@clarusway:~$ echo This command line \
> > is split in three \
> > parts
ubuntu@clarusway:~$ This command line is split in three parts
```

sed komutu:

text içeriğindeki kelimeleri değiştirmek için kullanabiliriz:

önce sed-awk-command isminde bir dosya oluşturarak içinde, sed.txt dosyasını açalım ve metni içine yazalım:

```
ubuntu@ip-172-31-89-213:~$ mkdir sed-awk-command && cd sed-awk-command
ubuntu@ip-172-31-89-213:~/sed-awk-command$ vim sed.txt
ubuntu@ip-172-31-89-213:~/sed-awk-command$ sed 's/linux/ubuntu/' sed.txt
```

sed komutunu yazalım:

```
ubuntu@ip-172-31-89-213:~/sed-awk-command$ sed 's/linux/ubuntu/' sed.txt
Linux is an OS. Linux is life. Linux is a concept.
I like ubuntu. You like linux. Everyone likes linux.
Linux is free. Linux is good. Linux is hope.
```

ilk linux u değiřtirdi diğerklerine bakmadı.

3. yü değiřtirmek için:

```
ubuntu@ip-172-31-89-213:~/sed-awk-command$ sed 's/linux/ubuntu/3' sed.txt
Linux is an OS. Linux is life. Linux is a concept.
I like linux. You like linux. Everyone likes ubuntu.
Linux is free. Linux is good. Linux is hope.
```

hepsini değiřtirmek için:

```
ubuntu@ip-172-31-89-213:~/sed-awk-command$ sed 's/linux/ubuntu/g' sed.txt
Linux is an OS. Linux is life. Linux is a concept.
I like ubuntu. You like ubuntu. Everyone likes ubuntu.
Linux is free. Linux is good. Linux is hope.
```

case sensitive olmadan tümünü değiřtirmek için:

```
ubuntu@ip-172-31-89-213:~/sed-awk-command$ sed 's/linux/ubuntu/ig' sed.txt
ubuntu is an OS. ubuntu is life. ubuntu is a concept.
I like ubuntu. You like ubuntu. Everyone likes ubuntu.
ubuntu is free. ubuntu is good. ubuntu is hope.
```

bütün değışiklikleri on the fly yani memory de tutarak bize gösterir. yazdırmak için:

```
ubuntu@ip-172-31-89-213:~/sed-awk-command$ sed 's/linux/ubuntu/ig' sed.txt >> new.txt
ubuntu@ip-172-31-89-213:~/sed-awk-command$ cat new.txt
ubuntu is an OS. ubuntu is life. ubuntu is a concept.
I like ubuntu. You like ubuntu. Everyone likes ubuntu.
ubuntu is free. ubuntu is good. ubuntu is hope.
```

new.txt ye yazdırdık.

örneğin 1.yi atlayarak 2. den itibaren hepsini değiřtirmek için:

```
ubuntu@ip-172-31-89-213:~/sed-awk-command$ sed 's/linux/ubuntu/2ig' sed.txt
Linux is an OS. ubuntu is life. ubuntu is a concept.
I like linux. You like ubuntu. Everyone likes ubuntu.
Linux is free. ubuntu is good. ubuntu is hope.
```

tek bir satırı mesela 2. satırın tamamını değiřtirmek için:

```
ubuntu@ip-172-31-89-213:~/sed-awk-command$ sed '2 s/linux/ubuntu/ig' sed.txt
Linux is an OS. Linux is life. Linux is a concept.
I like ubuntu. You like ubuntu. Everyone likes ubuntu.
Linux is free. Linux is good. Linux is hope.
```

sadece 2.satırda 2.den başlayarak tamamını değiřtirmek için:

```
ubuntu@ip-172-31-89-213:~/sed-awk-command$ sed '2 s/linux/ubuntu/2ig' sed.txt
Linux is an OS. Linux is life. Linux is a concept.
I like linux. You like ubuntu. Everyone likes ubuntu.
Linux is free. Linux is good. Linux is hope.
```

awk komutu:

Aho, Weinberger & Kernighan isimli kişilerin oluşturduğu komut.

awk.txt dosyası oluşturup içine şunları ekleyelim:

```
ubuntu@ip-172-31-89-213:~/sed-awk-command$ vim awk.txt
ubuntu@ip-172-31-89-213:~/sed-awk-command$ awk '{print}' awk.txt
This is line 1
This is line 2
This is line 3
This is line 4
This is line 5
```

print ile kullanırsak dosyanın çıktısını verir:

```
ubuntu@ip-172-31-89-213:~/sed-awk-command$ awk '{print}' awk.txt
This is line 1
This is line 2
This is line 3
This is line 4
This is line 5
```

sütunu yazdırabiliriz:

```
ubuntu@ip-172-31-89-213:~/sed-awk-command$ awk '{print $2}' awk.txt
is
is
is
is
is
```

bir çok sütunu yazdırabiliriz:

```
ubuntu@ip-172-31-89-213:~/sed-awk-command$ awk '{print $4, $2, $3, $1}' awk.txt
1 is line This
2 is line This
3 is line This
4 is line This
5 is line This
```

awk.txt yi şu şekilde güncelliyoruz:

```
awk.txt
This is part 1 of line 1 : This is part 2 of line 1
This is part 1 of line 2 : This is part 2 of line 2
This is part 1 of line 3 : This is part 2 of line 3
This is part 1 of line 4 : This is part 2 of line 4
This is part 1 of line 5 : This is part 2 of line 5
...
```

```
ubuntu@ip-172-31-89-213:~/sed-awk-command$ awk -F: '{print $2}' awk.txt
This is part 2 of line 1
This is part 2 of line 2
This is part 2 of line 3
This is part 2 of line 4
This is part 2 of line 5
```

Yukarıda -F: dediğimizde bu metni : ile ayrılan iki sütun olarak düşün  
{print \$2} dile de 2.sütunu yazdır.

Aşağıda yine -F: ile : ayraç olarak verdik. 1. sütunu yazdır, 1. sütunu içinde de 2. sütunu yazdır:

```
ubuntu@ip-172-31-92-42:~/sed-awk-command$ awk -F: '{print $1}' awk.txt | awk '{print $2}'
is
is
is
is
is
```

bir başka şekilde kullanım:

```
ubuntu@ip-172-31-89-213:~/sed-awk-command$ ls -l
total 12
-rw-rw-r-- 1 ubuntu ubuntu 260 Dec 14 20:57 awk.txt
-rw-rw-r-- 1 ubuntu ubuntu 157 Dec 14 20:47 new.txt
-rw-rw-r-- 1 ubuntu ubuntu 148 Dec 14 20:42 sed.txt
ubuntu@ip-172-31-89-213:~/sed-awk-command$ ls -l | awk '{print $9}'
awk.txt
new.txt
sed.txt
```

awk içine if de kullanalım:

```
ubuntu@ip-172-31-89-213:~/sed-awk-command$ awk '{ if($7 == "3") print $0;}' awk.txt
This is part 1 of line 3 : This is part 2 of line 3
ubuntu@ip-172-31-89-213:~/sed-awk-command$
```

awk.txt içinde 7.sütun 3 eşit ise \$0 yani o satırın tamamını yazdır.

crontab komutu:

DevOps kısmında sistem otomasyonu için kullanacağımız bir komut olacak:

cron table ın kısaltmasıdır. 7/24 çalışan bir server a düzenli aralıklarla çalıştırmak istediğimiz komutları bu kodla giriyoruz. Ya da gece yarısında bir görev planlamak için.

```
crontab -e          # edit the crontab file
crontab -l          # list current cron tasks
crontab -u username -e # edit other users's crontab file
...
```

crontab -e ile crontab dosyamızı editleyebiliriz.

crontab -l ile mevcuttaki cron görevlerini listele

crontab -u username -e ile başka kullanıcının crontab dosyasını editler

```
* * * * * <shell command> # execute cron job every minute
```

```
0 1 * * * <shell command> # execute cron job every day at 1 a.m.  
* * 1 * * <shell command> # execute every minute in January  
* * * * 6 <shell command> # execute every minute on every saturday  
0 1/15 * jan,jun mon,fri <command> # execute at every 1 a.m. and 3  
p.m. every monday and friday on  
january and june
```

sırasıyla;  
her dakika  
her gün 1 am de  
Ocak ayının her dakikasında  
Her cumartesi her dakika  
Ocak ve Haziran aylarının Her pazartesi ve cuma günlerinde her 1  
a.m ve 3 p.m de

crontab -e diyerek editörü açıp  
her dakika çalışacak iki komutu ekledik:

```
***** cp /etc/passwd ~/passwd_backup$date.txt  
***** date >> /home/ubuntu/date.log
```

ikinci satırdaki komutta tarihi ve saati her dakika date.log  
dosyasına otomatik yazıyor:

```
date.log  
1 Wed Dec 14 21:11:01 UTC 2022  
2 Wed Dec 14 21:12:01 UTC 2022  
3 Wed Dec 14 21:13:01 UTC 2022  
4 Wed Dec 14 21:14:01 UTC 2022  
5
```

<https://crontab.guru/>