

## 22.Doküman

## Zamanlanmış Görevler

Konu Etiketleri

zamanlanmış görevler , cron , crontab

## Zamanlanmış Görevler

Sistemde yapılması gereken rutin işlerin zamanı geldiğinde otomatik olarak yapılması işine zamanlanmış görevler deniyor. Bu rutin işlere örnek söylemek gerekirse; benim her pazartesi sistemi yedeklemem gerekiyor diyelim, bunu her pazartesi elle yapmak yerine bu işi zamanlanmış görevlere ekleyerek otomatiğe bağlayabilirim. İşte zamanlanmış görevler bu ve bunun gibi durumlarda sıkça kullanılıyor.

## cron

Rutin tekrarları sağlayan zamanlanmış görevleri yerine getirmemizi sağlayan servisimizin adı **cron**'dur.

Anlatıma **cron** servisinin çalışma durumunu kontrol ederek başlayalım. Bunun için konsola **service cron status** komutunu verelim.

```
root@taylan:~# service cron status
● cron.service - Regular background program processing daemon
   Loaded: loaded (/lib/systemd/system/cron.service; enabled; vendor preset: en
   Active: active (running) since Tue 2020-01-14 11:41:11 EST; 1min 21s ago
     Docs: man:cron(8)
  Main PID: 508 (cron)
    Tasks: 1 (limit: 4915)
   Memory: 608.0K
    CGroup: /system.slice/cron.service
            └─508 /usr/sbin/cron -f

Oca 14 11:41:11 taylan systemd[1]: Started Regular background program processin
Oca 14 11:41:12 taylan cron[508]: (CRON) INFO (pidfile fd = 3)
Oca 14 11:41:12 taylan cron[508]: (CRON) INFO (Running @reboot jobs)
lines 1-13...skipping...
```

Cron servisimizin çalıştığını teyit etmiş olduk. Cron servisinin yapılandırma dosyası **/etc/crontab** konumunda yer alıyor göz atmak için konsola **cat /etc/crontab** komutunu verebiliriz.

```
root@taylan:~# cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab`
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# Example of job definition:
# .----- minute (0 - 59)
# | .----- hour (0 - 23)
# | | .----- day of month (1 - 31)
# | | | .----- month (1 - 12) OR jan,feb,mar,apr ...
# | | | | .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# | | | | |
# * * * * * user-name command to be executed
17 * * * * root    cd / && run-parts --report /etc/cron.hourly
25 6 * * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6 * * 7 root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6 1 * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
#
```

Şimdi de düzenleme yapmak adına konsola **crontab -e** komutunu verelim. Burada yer alan **-e** parametresi **"edit"** kelimesini yani **"düzenleme"** ifadesini temsil ediyor.

```

root@taylan:~# crontab -e
no crontab for root - using an empty one

Select an editor. To change later, run 'select-editor'.
 1. /usr/bin/vim.gtk
 2. /bin/nano          <---- easiest
 3. /usr/bin/vim.basic
 4. /usr/bin/mcedit
 5. /usr/bin/vim.tiny

Choose 1-5 [2]: 2

```

Konsol bize düzenleme yaparken hangi aracı kullanmak istediğimizi soruyor. Ben **nano** aracını seçiyorum yani konsola **2** rakamını giriyorum.

```

GNU nano 2.9.1 /tmp/crontab.fLrmb3/crontab

# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)

^G Get Help      ^O Write Out    ^W Where Is     ^K Cut Text     ^J Justify      ^C Cur Pos
^X Exit          ^R Read File    ^\ Replace      ^U Uncut Text   ^T To Spell     ^  Go To Line

```

Çıktıda bize bir örnek kullanım gösterilmiş biz bu kullanımı ele alarak servisin kullanım şeklini görelim.

```

#İş tanımı örneği:
.----- dakika (0 - 59)
| .----- saat (0 - 23)
| | .----- ayın günü (1 - 31)
| | | .----- ay (1 - 12) OR ocak,şubat,mart,nisan ...
| | | | .----- haftanın günleri (0 - 7) (Pazar= 0 ya da 7, Pazartesi=1, Salı=2 .. , Cumartesi=6)
| | | | | .----- kullanıcı adı
| | | | | | .-- komut
| | | | | |
* * * * *
0 5 * * 1 root tar -zcf /var/backups/home.tgz /home/

```

Yukarıdaki şema yeterince açıklayıcı olmadıysa endişelenmenize gerek yok. Örnekler üzerinden yapacağımız anlatım ile **cron** servisimizin kullanımını rahat şekilde anlamış olacağınıza inanıyorum.

Zamanlanmış görevleri yerine getiren servis **cron** ve bu servisin yerine getirdiği işlere de **cron job** deniliyor. Bu tanımın çok da önemi yok ancak bunu böyle bilerseniz daha iyi olur. Bizler de kendi zamanlanmış görevlerimizi (cron jobs) atamak istersek **crontab -e** komutu ile açtığımız dosyanın son satırına yapmak istediğimiz işi ve tam tarih

düzenini girmemiz gerekiyor. Bu görevleri nasıl oluşturacağımızı konunun devamında örnekler üzerinden öğreniyor olacağız.

## Örnekler

Kullanımın anlaşılması amacıyla örnekleri olabildiğince basit tuttum.

Sizlere vereceğim örnekler ne kadar gereksiz gelse de örneği konuyu anlamak adına kullandığımızı unutmayın ve asıl anlatılmak istenen noktaya odaklanın lütfen.

O anın tarihi detaylarını bir dosyaya kaydetmek istiyorum diyelim. Tarihi detayları **zamanlar.txt** isimli klasöre kaydetmek için `* * * * * echo `date` ~/Desktop/zamanlar.txt` komutunu kullanacağım.

Bütün yıl boyunca her dakikanın çıktılarını kaydetmesini istersem aşağıdaki şekilde bir kullanım işimizi görür.

The image shows two windows. The top window is a terminal running GNU nano 2.8.7, editing the crontab file at /tmp/crontab.xA7C62/crontab. The file content is as follows:

```
GNU nano 2.8.7 Dosya: /tmp/crontab.xA7C62/crontab Değiştirildi

# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
* * * * * echo `date` >> ~/Desktop/zamanlar.txt
```

The bottom window is a terminal showing the output of the crontab command:

```
DAKİKA  SAAT  AYIN GÜNLERİ  AYLAR  HAFTANIN GÜNLERİ  KULLANICI ADI  KOMUT
*      *      *      *      *      *      *      echo `date` >> ~/Desktop/zamanlar.txt
```

Yıldız işareti (\*) olan kısımlar hepsini kapsamaktadır yani haftanın bütün günleri, ayın bütün günleri ve saatin bütün dakikaları.. gibi.

Komutun düzenli hali `* * * * * echo `date` ~/Desktop/zamanlar.txt` şeklindedir.

Bir süre geçtikten sonra masaüstünde(Desktop dizininde) yer alan **zamanlar.txt** dosyasını `cat zamanlar.txt` komutu ile kontrol ediyorum.

```

root@taylan:~# cat zamanlar.txt
Sal Oca 14 11:44:59 EST 2020
Sal Oca 14 11:45:59 EST 2020
Sal Oca 14 11:46:59 EST 2020
Sal Oca 14 11:47:59 EST 2020
Sal Oca 14 11:48:59 EST 2020
Sal Oca 14 11:49:59 EST 2020
Sal Oca 14 11:50:59 EST 2020
Sal Oca 14 11:51:59 EST 2020
root@taylan:~#

```

Görev istediğimiz şekilde yani her dakikada yazılacak şekilde otomatik olarak gerçekleşmiş oldu.

Şimdi spesifik bir zaman belirleyerek görevin yerine getirilme durumunu ele alalım.

Her gün 10:25 da otomatik olarak **test.txt** isimli belgeye "test yazısı" yazdırmak için kalıbı **crontab** dosyasına aşağıdaki şekilde yazmalıyız.(Ayrıca başta hangi ifadenin neyin karşılığı olduğunu belirttiğim için tekrar tekrar üzerlerine yazmıyorum. Eğer şaşırırsanız yukarıdaki kısımları tekrar gözden geçirebilirsiniz.)

```

25 10 * * * echo "test yazısı" ~/Desktop/test.txt
GNU nano 2.8.7 Dosya: /tmp/crontab.nmcAtJ/crontab Değiştirildi

# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
* * * * * echo `date` >> ~/Desktop/zamanlar.txt
25 10 * * * echo "test yazısı" >> ~/Desktop/test.txt

```

<sup>^</sup>G Yardım Al <sup>^</sup>O Yaz <sup>^</sup>W Ara <sup>^</sup>K Metni Kes <sup>^</sup>J Yasla <sup>^</sup>C İmleç Pozisyonu  
<sup>^</sup>X Çık <sup>^</sup>R Dosya Oku <sup>^</sup>\ Değiştir <sup>^</sup>U Metni Kesme <sup>^</sup>T Denetime <sup>^</sup> Satıra Git

**ls -l** komutu ile de dosyamızın üretildiği tarihi sorguladık ve çıktıdan da anladığımız üzere verdiğimiz görev zamanında yerine getirilmiş.



```

root@taylan:~/Desktop# ls -l
total 8688
-rw-r--r-- 1 root root 4379880 Feb 15 2016 aircrack
-rw-r--r-- 1 root root 4379880 Feb 15 2016 aircrack-ng-1.2-rc4.tar.gz
----- 1 root root 4755 Jan 10 14:34 filazof
----- 1 root root 42 Jan 9 19:08 isimler.txt
----- 1 root root 214 Jan 10 22:39 isimler.zip
----- 1 root root 40 Jan 9 19:22 'isimler_kopya (kopya).txt'
----- 1 root root 40 Jan 9 19:22 isimler_kopya.txt
----- 1 root root 66 Jan 10 11:03 kelimeler.txt.gz
----- 1 root root 228 Jan 9 17:07 liste
----- 1 root root 228 Jan 9 17:11 liste_bilgisi
----- 1 root root 1221 Jan 10 20:48 metin.tar.bz2
----- 1 root root 486 Jan 10 20:47 metin.tar.gz
d----- 4 root root 4096 Jan 10 21:37 metin_dosyalari
-rw-r--r-- 1 root root 11 Jan 18 10:25 test.txt ←
----- 1 root root 56 Jan 10 16:53 test.txt.bz2
d----- 3 root root 4096 Jan 11 12:36 yeni_dizin
----- 1 root root 54012 Jan 16 14:51 zamanlar.txt

```

Eğer belirli bir saat-tarih arasında belirli kere tekrar eden görev atamak istersek kullanım şekilleri aşağıdaki örneklerdeki gibi olacaktır.

Belirli aralıklar ile olacaksa aralıkları **-(kısa çizgi)** işareti ile ayırarak belirtmemiz gerekiyor. Yani örneğin;

Her saatin ilk 5 ile 20 dakikası arasında komutum çalışsın istersem:

```
5-20 * * * * komut
```

Her gün saat 3 ile 5 arasında her dakika komutum çalışsın istersem:

```
* 3-5 * * * komut
```

Her ayın 5 ile 15. günleri arasında saat 16 ile 22 arasındaki her saatin ilk 20 dakikası komutum çalışsın istersem.

```
0-20 16-22 5-15 * * komut
```

Haziran ayında ilk 20 gününün her cuma günü saat 12 ile 15 arasındaki 35 ile 55 dakikaları arasında komutum çalışsın istersem:

```
35-55 12-15 1-20 6 5 komut
```

Belirli gün ve tarihlerde olacaksa o tarihleri **,(virgül)** işareti ile ayırarak tek tek belirtmemiz gerekiyor. Yani örneğin;

Her saatin 5 10 15 ve 20. dakikalarında komutum çalışsın istersem:

```
5,10,15,20 * * * * komut
```

Her gün saat 3 5 ve 7 de komutum çalışsın istersem:

```
0 3,5,7 * * * komut
```

Her ayın 20 22 24 ve 26. günleri 2 4 6 ve 8 saatlerinin 5. ve 10. dakikalarında komutum çalışsın istersem:

```
5,10 2,4,6,8 20,22,24,26 * * komut
```

Belirli süre içinde belirli kez çalışmasını istersek **/(taksim)** işareti ile kaç kez olacağını belirtebiliyoruz. Yani örneğin;

5 dakikada bir çalışmasını istersem:

```
*/5 * * * * komut
```

10 saatte bir komutumun çalışmasını istersem:

```
* */10 * * * komut
```

Elbette örnekleri çoğaltmak mümkün ancak daha fazlası gereksiz olacaktır. Ayrıca atadığınız zamanlanmış görevlerin doğruluğunu kontrol etmek isterseniz [bu adresten](#) verdiğiniz zaman aralığının karşılığını görebilir ve hata yapmaktan kaçınabilirsiniz.

Yani bu konu sizlerin de biraz kurcaladıktan sonra anlayacağı kolay bir konudur.

Eğer belirlenen zamanlanmış görevleri listelemek istersek `crontab -l` komutunu kullanabiliriz. Buradaki `-l` parametresi **"list"** kelimesini yani **"listeleme"** işlevini temsil ediyor.

```
root@taylan:~# crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
25 10 * * * echo "test yazısı" ~/Desktop/test.txt
```

Şayet oluşturduğumuz tüm zamanlanmış görevleri silmek istersek `crontab -r` komutunu kullanmalıyız. Buradaki `-r` parametresi **"remove"** kelimesini yani **"silme"** işlevini temsil ediyor.

```
root@taylan:~# crontab -r
root@taylan:~# crontab -l
no crontab for root
root@taylan:~#
```

### Alıştırmalar Hakkında

Yalnızca okumak yetmez, öğrendiğiniz bilgilerin kalıcı olabilmesi için bolca alıştırma yapmalısınız. Doküman içerisindeki bilgileri pekiştirmek için aşağıdaki alıştırmalar ile başlayabilirsiniz. Elbette burada yer alan alıştırma faaliyetleri dışında, konuyu öğrendiğinizi hissette kadar kendiniz de bolca pratik yapmayı da ihmal etmeyin lütfen. Aksi halde öğrendiğiniz bilgiler kısa sürede unutulup gidecektir.

Her gün 11:11'de çalışan bir görev atayın.

Her pazartesi 12:00'da çalışan bir görev atayın.

Her ayın 10'u ile 25'i arasında her gün saat 11 ile 22 arasındaki her saatin ilk 10 dakikası çalışan bir görev atayın.

Her saatin 1 3 5 ve 7 dakikalarında çalışan bir görev atayın.

Her gün tam olarak saat 2 5 ve 9 da çalışan bir görev atayın.

Her ayın 10 12 14 ve 26. günleri 2 3 6 ve 22 saatlerinin 3. ve 15. dakikalarında çalışan bir görev atayın.

Mart ayının ilk 18 gününün her pazar günü saat 14 ile 19 arasındaki 15 ile 35 dakikaları arasında çalışan bir görev atayın.

2 saate bir çalışan bir görev atayın.

8 dakikada bir çalışan bir görev atayın.

İlgili komut yardımıyla atadığınız görevleri listeleyin.

İlgili komut yardımıyla tüm zamanlanmış görevleri silin.

## Geri Bildirimde Bulunun

Sizlere daha verimli bir kaynak sunabilmemiz için, uygulamada veya dokümantasyonlarda yer alan tüm hata ve eksiklerimizi bize bildirebilirsiniz.

Geri Bildirimde Bulunun