

1. Cron Nedir?

Cron, Linux sistemlerinde zamanlanmış görevleri otomatikleştirmek için kullanılan bir zamanlayıcı hizmetidir. Kullanıcıların belirli zaman aralıklarında komutları çalıştırmasına veya betikleri yürütmesine olanak tanır. Cron hizmeti, sürekli çalışan bir "cron daemon" (crond) ile desteklenir. Bu hizmet, hem kullanıcıların hem de sistem yöneticilerinin, sistem üzerinde düzenli işlemleri otomatik hale getirmesine olanak sağlar. Örneğin, yedekleme işlemleri, sistem temizliği veya log yönetimi gibi görevler cron kullanılarak kolaylıkla otomatikleştirilebilir.

2. Cron Tabloları (Crontab)

Crontab ("cron table"), kullanıcıların zamanlanmış görevleri tanımladığı bir dosyadır. Bu dosya, her kullanıcı için ayrı olarak oluşturulur ve zamanlama kurallarıyla birlikte hangi komutların ne zaman çalıştırılacağını belirtir.

Crontab Dosyası Formatı

Her bir satır, bir cron görevini temsil eder ve aşağıdaki formatı kullanır:

```
\ \ \

* * * * * komut

-----

| | | |
| | | +----- Haftanın günü (0 - Pazar, 6 - Cumartesi)
| | | +----- Ay (1 - 12)
| | +----- Ayın günü (1 - 31)
| +----- Saat (0 - 23)
+----- Dakika (0 - 59)
\ \ \
```

Crontab Kısayolları

Cron zamanlama formatı için sıkça kullanılan kısayollar:

- `@reboot` : Sistem başladığında bir kez çalışır.
- `@hourly` : Her saat başında çalışır.
- `@daily` veya `@midnight` : Her gece yarısında çalışır.
- `@weekly` : Her hafta başında çalışır.
- `@monthly` : Her ay başında çalışır.
- `@yearly` veya `@annually` : Her yıl başında çalışır.

Crontab Komutları

- ``crontab -e`` : Crontab dosyasını düzenlemek için kullanılır.
- ``crontab -l`` : Mevcut crontab görevlerini listelemek için kullanılır.
- ``crontab -r`` : Tüm crontab görevlerini silmek için kullanılır.
- ``crontab -u [kullanıcı_adı]`` : Belirli bir kullanıcının crontab'ını düzenlemek veya görüntülemek için kullanılır (yönetici yetkisi gerektirir).

Crontab düzenlenirken, varsayılan metin düzenleyici genellikle ``nano`` olarak atanır. Ancak, ``EDITOR`` değişkeni kullanılarak bu ayar değiştirilebilir. Örneğin, ``export EDITOR=vim`` komutuyla düzenleyici ``vim`` olarak ayarlanabilir.

3. Cron Kullanımı Örnekleri

Örnek 1: Her Saat Başında Bir Komut Çalıştırma

Her saat sistemin disk kullanımını kontrol eden bir betiği çalıştırın:

```
$ 0 * * * * /path/to/check_disk_usage.sh
```

Bu komut, saat başında ``check_disk_usage.sh`` betiğini çalıştırarak disk kullanımını kontrol eder ve raporlar.

Örnek 2: Belirli Bir Gün ve Saatte Komut Çalıştırma

Her ayın 1'inde saat 03:00'a sistem yedeği alın:

```
$ 0 3 1 * * /path/to/backup_script.sh
```

Bu görev, sistemin yedekleme işlemlerini düzenli ve otomatik bir şekilde gerçekleştirmesini sağlar.

Örnek 3: Haftada Bir Log Temizleme

Her Pazartesi sabahı 04:00'a log dosyalarını temizleyin:

```
$ 0 4 * * 1 /path/to/clear_logs.sh
```

Bu görev, log dosyalarının birikmesini önler ve disk alanını optimize eder.

Örnek 4: 10 Dakikada Bir Komut Çalıştırma

10 dakikada bir belirli bir URL'yi kontrol eden bir betiği çalıştırın:

```
$ */10 * * * * /path/to/check_url.sh
```

Bu tür bir zamanlama, özellikle ağ durumu veya hizmet sürekliliğini izlemek için kullanışlıdır.

Örnek 5: Yeniden Başlatıldığında Bir Komut Çalıştırma

Sistem yeniden başlatıldığında bir dosyaya bilgi kaydedin:

```
$ @reboot echo "Sistem yeniden başlatıldı: $(date)" >> /var/log/reboot.log
```

Bu komut, sistem yeniden başlatıldığında zaman damgalı bir mesaj ekler ve yeniden başlatma geçmişinin izlenmesini sağlar.

4. Cron Loglama

Cron görevlerinin başarılı şekilde çalışıp çalışmadığını anlamak için loglar kontrol edilebilir:

- Sistem log dosyası: `/var/log/syslog` (Bazı Linux dağıtımlarında `cron` logları buradadır.)

```
$ grep CRON /var/log/syslog
```

- Cron logları, hata durumlarında detay verir.

Hata durumlarını çözmek için loglardaki bilgilerden yararlanabilirsiniz. Örneğin, bir görev çalışmadığında cron loglarına bakarak, sorunun nedenini hızlıca tespit edebilirsiniz.

5. Sık Karşılaşılan Sorunlar ve Çözümler

Sorun 1: Cron Görevi Çalışmıyor

- Sebep: Betik veya komutun yolu yanlış olabilir.

- Çözüm: Komutlarda tam yolları kullanın. Örneğin:

```
$ /usr/bin/python
```

Sorun 2: Cron Çıktısı Nerede?

- Sebep: Cron varsayılan olarak çıktıyı sessizce yönetir.

- Çözüm: Komutun çıktısını bir dosyaya yönlendirin:

```
$ /path/to/command > /path/to/output.log 2>&1
```