

Upstart

Çağrı Emer

Nisan, 2016

İçindekiler

1	Giriş	2
2	Temel Kısımlar	4
3	Örnek Uygulama	5
4	Kullanıcı Süreçleri	6
5	Kaynak:	7

1 Giriş

6.10'dan beri Ubuntu'da olan Upstart, init artsürecinin yerini alan bir uygulamadır. Uygulama, farklı dağıtımlarda da yer almaktadır.

Upstart işletim sisteminin başlatılması esnasında görev ve servislerin başlatılması, kapatılması esnasında durdurulması, bilgisayar çalışırken de kontrol edilmesi işlevini yerine getiren ve 6.10'dan itibaren Ubuntu'da geleneksel init artsürecinin yerini alan uygulamadır. Aslında Ubuntu için geliştirilmesine rağmen, init kullanan tüm Linux dağıtımlarında geleneksel init'in yerini alabilir. Şu anda bilinen kullanıcıları arasında Ubuntu dışında Debian, Red Hat, Google Chrome OS ve Maemo gibi dağıtımlar bulunmaktadır.

Upstart betiklerinin nasıl yazılacağına geçmeden önce init ne iş yapar değinmek doğru olacaktır. Çekirdek tarafından başlatılan init, sistemdeki bütün süreçlerin başlatılması işlevini yerine getirir. init tarafından yönetilen tüm süreçler "iş" olarak tanımlanır ve /etc/init dizini altındaki dosyalarda tutulur. Ubuntu'daki init(8)* olay-tabanlı bir artsüreçtir. Bu da sistem durumundaki herhangi bir değişiklikte işlerin ya da süreçlerin otomatik olarak başlatılıp durdurulabilmesini sağlar. Bağımlılık-tabanlı init artsüreçlerinden farkları bu işlemleri yapmak için başka işlerin başlamasını beklememek ve bağımlılık listelerini bir döngüyle taramamaktır denebilir.

Sistemde init ile ilgili kullanılabilecek üç dizinden söz edilebilir. Bunlar şöyledir:

Dizinler	Açıklama
/etc/init	Upstart yapılandırma dosyalarının tutulduğu dizindir.
/etc/init.d	Geleneksel init betiklerinin ve geriye dönük uyumluluğa sahip Upstart yapılandırmalarının bulunduğu dizindir.
/etc/default	Upstart ve geleneksel init'in çalışma şekillerini kontrol eden dosyaların tutulduğu dizindir.

Sayılan dizinlerde tutulan dosyalar özetle, bir süreci kontrol etmek için yapılması gerekli olan işlemleri barındırır. Upstart sayesinde örneğin SSH sunucuyu şu komutlarla yönetmek mümkündür:

Komut	İşlevi
start ssh	Süreci başlatır.
stop ssh	Süreci sonlandırır.
reload ssh	Sürecin yeni yapılandırma dosyalarını kullanmasını sağlar.

Komut	İşlevi
restart ssh	Sürece SIGHUP sinyali göndererek yeniden başlatılmasını sağlar.
status ssh	Sürecinin çalışıp çalışmadığı bilgisini döndürür.

Bütün bu yukarıda sayılan iş tanımları Upstart sayesinde mümkündür. `/etc/init/ssh` dosyası içerisinde ssh artsürecini başlatan `exec` kısmı bulunur. Bir uygulamanın var olan init dosyasını incelemektense, sıfırdan bir init dosyası örneği oluşturmak konseptin anlaşılması adına daha iyi olacaktır. Ubuntu’da bu işi sıfırdan yapmak için `init(5)**` man sayfası çok iyi bir kaynaktır. Bu dosya oluşturulacak betiklerin `/etc/init` dizininde `.conf` ya da `.override` uzantısı ile oluşturulması gerektiğini söyler. `.conf` uygulamanın ya da sürecin temel yapılandırması iken `.override` temel yapılandırmaki tanımları ezen kuralları içeren dosyadır. Dolayısıyla bir süreç için yalnızca bir `.conf` dosyası ya da hem `.conf` hem de `.override` dosyası kullanılabilecekken, yalnızca `.override` dosyasının kullanılması mümkün değildir. Yapılandırma dosyaları çalıştırma izinleri olmayan düz metin dosyaları olmalıdır.

2 Temel Kısımlar

exec - script

Bütün iş dosyaları exec ya da script kısmı içermek zorundadır. Fakat ikisini birden barındıramazlar. ¹ Bu iki kısmın farkı, exec ile bir komut verilebilecekken script ile komutlardan oluşan bir betik yazılabilmektedir. Bir de eğer “script” kullanıldıysa “end script” şeklinde sonlandırmak gerekmektedir.

pre-start - post-start - pre-stop - post-stop

Çalıştırmak istenilen işi başlatmadan ya da durdurmadan önce ve başlattıktan ya da durduktan sonra çalıştırılacak komutlar da pre/post-start/stop script kısımları altında tanımlanmaktadır. Yine bu kısımlar “end script” ile sonlandırılır.

start on - stop on

Yapılandırma dosyasına eklenecek bu satırlar hangi olay sonucu sürecin başlatılacağını ya da durdurulacağını belirtir. Bu kısımda tanımlanabilecek ilk olay startup’tır. Bilgisayar ilk başladığında dosya sistemi sadece okunabilir hâldeyken ve ağ bağlantılarının olmadığı durumu belirtir. Belirli işleri belirli çalışma düzeylerinde (runlevel) başlatmak için kullanılacak seçenek runlevel’dır. Bahsedilmeye değer kullanılacak son iki seçenek ise started ve stopped seçenekleridir. Bu seçenekler kontrol edilmek istenen süreçlerin hangi süreçler başlatıldıktan ya da durdurulduktan sonra yönetileceğini anlatır. Ubuntu için tanımlı bütün olayların listesine Upstart Wiki’sinde bulunan “cookbok” isimli belgenin ilgili kısmından ulaşılabilir. ²

console

Bu yazıda değinilecek son önemli kısım console kısmıdır. Console kısmı sayesinde bir sürecin girdi ya da çıktıların nereden alınıp nereye gönderileceği kontrol edilebilir. Kullanılacak dört seçenek logged, output, owner ve none seçenekleridir. none için /dev/null kullanılır ve süreçle etkileşime girilmez. output seçeneği /dev/console’u kullanır ve işleri uçbirim üzerinden yürütür. owner seçeneği output’a ek olarak uçbirimin sahibi kullanıcıya süreçlere CTRL+C gibi belirli sinyaller gönderebilme olanağı tanır. Öntanımlı olarak gelen logged ise çıktıların log tutucuya gönderileceğini belirtir.

Bu yazıda bahsedilen kısımlar dışında kullanılacak kısımların tam listesine Upstart Wiki’den ³ erişilebilir. Ayrıca detaylı bir başucu kaynağı upstart cookbook ⁴ da sık kullanılanlar arasında yerini almalıdır.

¹[http://upstart.ubuntu.com/wiki/Stanzas?highlight=\(\(CategoryDoc\)\)#script](http://upstart.ubuntu.com/wiki/Stanzas?highlight=((CategoryDoc))#script)

²<http://upstart.ubuntu.com/cookbook/#ubuntu-well-known-events-ubuntu-specific>

³http://upstart.ubuntu.com/wiki/Stanzas_for_0.5.1

⁴<http://upstart.ubuntu.com/cookbook/>

3 Örnek Uygulama

Şimdi yukarıda tanımlanan temel işlemleri kullanan iki basit yapılandırma dosyası oluşturulacaktır.

```
1 sudo nano /etc/init/surecbir.conf
```

komutu ile dosya açılarak içerisine şunlar yazılır:

```
1 pre-start script
2     mkdir /home/<kullanici_adi>/Masaüstü/dizinim
3 end script
4
5 script
6     echo "Çalışıyor!" > /home/<kullanici_adi>/Masaüstü/dizinim/calisma.log
7 end script
8
9 post-start script
10     sleep 5
11 end script
12
13 post-stop script
14     chown -R <kullanici_adi>:<kullanici_adi> /home/<kullanici_adi>/Masaüstü/dizinim
15 end script
```

CTRL+X ile dosya kaydedilir.

```
1 sudo nano /etc/init/sureciki.conf
```

komutu ile yeni bir dosya açılarak içerisine şunlar yazılır:

```
1 start on stopped surecbir
2
3 script
4     echo "Bu iş ilk işin sonlanması ile başladı." >> /home/<kullanici_adi>/Masaüstü/dizinim/
5     calisma.log
6 end script
```

CTRL+X ile dosya kaydedilir.

Şu anda iki adet süreci tanımlamış bulunmaktasınız. Bunlar “surecbir” ve “sureciki” olarak adlandırıldılar. İkinci süreç, ilk sürece bağımlı olduğundan onun elle başlatılmasına gerek yoktur. İlk süreci şu komut ile başlatabilirsiniz:

```
1 sudo service surecbir start
```

Bu komut sonrasında ilk olarak masaüstünde bir dizinim klasörü oluşturulacaktır. Tanımı pre-start altında yapılmıştır. Ardından dizinim içerisinde calisma.log isimli bir dosya oluşturulacak ve içeriğine “Çalışıyor!” yazılacaktır. Bu işlemin de tanımı script altında verilmiştir. Ardından surecbir isimli süreç beş saniye boyunca bir şey yapmadan bekleyecek ve sonlanacaktır. Bu kısım da post-start altında tanımlı bulunmaktadır. Sürecin sonlanması ile masaüstünde bulunan dizinim klasörünün sahipliği içindeki dosyalar ile birlikte kullanıcınıza verilecektir. Bu işlem de post-stop kısmında tanımlanmıştır.

surecbir sürecinin sonlanmasının ardından sureciki süreci başlayacak ve calisma.log dosyasına “Bu iş ilk işin sonlanması ile başladı.” satırını ekleyecektir. sureciki’nin ne zaman başlayacağını tanımlayan kısım start on kısmıdır.

4 Kullanıcı Süreçleri

Upstart'ın güncel sürümleri, sistem kullanıcılarının kendi işlerini tanımlamalarına da izin vermektedir. Böylelikle yönetici hakları gerekmez, kullanıcılar kendi süreçlerini yaratabilmekte ve çalıştırabilmektedirler. Sistem için kullanılacak stanza'ların (Bu yazıda "kısım" olarak adlandırıldılar.) tümü henüz kullanıcı tanımlı işler için tanımlanmamış olmasına rağmen, temel işleri görebilecek şekilde kullanıcı tanımlı Upstart yapılandırmaları oluşturulabilmektedir. Bunun için yapılması gereken ev dizininde .init isimli bir dizin oluşturup iş tanımlarını tutan .conf dosyalarını bu dizin altına koymaktır. Bir de Ubuntu Upstart öntanımlı olarak kullanıcı işlerinin çalıştırılmasına izin vermediği için /etc/dbus-1/system.d/Upstart.conf dosyası bir yedeği alınarak, Upstart'ın Launchpad sayfasından indirilen yapılandırma ile değiştirilmelidir. Ardından start-stop gibi komutlarla kullanıcı tanımlı işler çalıştırılabilir.

.init dizinini oluşturmak ve Upstart.conf dosyasını değiştirmek için izlenmesi gereken adımlar kısaca şöyledir. Eğer bu yapılandırma değiştirilmek istenirse yedek dosyasının geri döndürülmesi yeterli olacaktır.

```
1 mkdir ~/.init
2 cd ~
3 wget http://bazaar.launchpad.net/~upstart-devel/upstart/trunk/download/head:/upstart.conf
  -20080508231852-jw3hh1a1d02temj7-1/Upstart.conf
4 sudo mv /etc/dbus-1/system.d/Upstart.conf /etc/dbus-1/system.d/Upstart.conf.yedek
5 sudo mv Upstart.conf /etc/dbus-1/system.d/Upstart.conf
```

Bu yazıda Upstart gibi çok geniş bir konuya giriş yapılmış ve temel işlemlerin nasıl yapılacağı anlatılmıştır. Daha kapsamlı işlemler için kullanıcıların başvuru yapabilecekleri kaynaklar gösterilmiş ve iki basit örnekle kavramın anlatımı tamamlanmaya çalışılmıştır. Başlangıç süreçleri ile ilgili daha fazla bilgi edinmek için /etc/rc* dizinlerinin ne işe yaradığının, o dizinlerde bulunan dosyaların nasıl isimlendirildiğinin ve nasıl yazıldığının araştırılması bu yazıya tamamlayıcı olması açısından önerilmektedir.

Not:\

man 8 init (5. ve 8. bölümlerinde tanımlanan init farklıdır)\
man 5 init (5. ve 8. bölümlerinde tanımlanan init farklıdır)

5 Kaynak: