

Süreç Sonlandırma

Semetey Coşkun

Nisan, 2016

Konuya, başlıkta adı geçen “süreç” ile başlayalım. Süreç yani İngilizce adı ile process; çalışan dosyaların çalışır durumdaki haline verilen isim olarak tanımlanıyor. Yani daha açık olarak ve örnek üzerinden anlatacak olursak; Pidgin anında mesajlaşma aracını çalıştırmak için kullandığımız “çalışan” dosya “/usr/bin/pidgin”. Bu komutu uçbirimden verdiğimiz (programı menüden tıklayarak çalıştırdığımızda da arkada olan olay bu komutun verilmesi tabii) zaman Pidgin programı belleğe yüklenmiş oluyor ve bu belleğe yüklenmiş durumundan artık süreç (process) olarak bahsediyoruz.

Çalıştırdığımız Pidgin’i sonlandırmak için izleyeceğimiz bir kaç yol mevcut. Tabii Pidgin’i tıkladık, normal şartlar altında da bir kaç alternatifimiz mevcut; bildirim alanında sağ tıklayarak; çıkış diyebiliriz. Ya da program ana sayfası açıkken menülerden kişiler -> çıkış alternatifini kullanarak, bu süreci sonlandırabiliriz; ki zaten olağan durum bunu gerektirir. Bizim inceleyeceğimiz ise şu durum; bir süreci sonlandırırken süreçten cevap alınmaması durumunda izlememiz gereken yol.

Bir programı kapatmak istiyoruz ve bu nedenle yukarıdaki (metacity’den bahsediyorum) çarpı işaretine bastık. Fakat süreç yanıt vermiyor. Durum bu şekildeyken epey alternatifimiz mevcut. pkill/killall. Bu komutlar ile cevap vermeyen programımıza uçbirimden ulaşmamız ve sonlandırmamız mümkün. Örneğin uçbirimden aşağıdaki komutu vermemiz bir alternatif olabilir.

```
1 pkill pidgin
```

Ya da başka alternatif olarak

```
1 killall pidgin
```

Ek bilgi;

Bu komutlar ile bir süreci sonlandırmak zorunda değilsiniz tabii. Örneğin kendi yazdığınız bir uygulama için yanıt vermeme durumlarında yapılacak işlemler için bir takım fonksiyonları devreye sokmak isteyebilirsiniz. “killall programadı” bu komut varsayılan olarak süreci sonlandırır. Fakat ismindeki kill sizi yanıltmasın, bu uygulama ile, önceden de belirttiğim gibi bir süreci sonlandırmak (kill) zorunda değilsiniz. killall komutunun şu kullanımını sıkça görebilirsiniz;

```
1 killall -9 program_adı
```

Bu komut ile yapılan şu; programa kill sinyali (SIGKILL) gönderiliyor. Bu sinyal ise programı doğrudan sonlandırmak için kullanılıyor. Fakat parametresini değiştirerek bu durum farklı bir hale getirilebilir. Örneğin;

```
1 killall -10 program_adı
```

Bu şekilde bir programa SIGKILL değil, SIGUSR1 sinyali göndermiş olursunuz. SIGUSR1 sinyali, üzerine fonksiyon yazılabilen bir sinyaldir. Yani siz programınıza şunu yaptırabilirsiniz;

SIGUSR1 sinyalini alırsan; “şu, şu, şu” fonksiyonları çalıştır. Ya da aynı durum SIGUSR2 ya da diğer sinyal fonksiyonu yazılabilen sinyaller için de geçerlidir.

```
1 killall -12 program_adı
```

Sürece SIGUSR2 sinyali gönderilir. Yani gördüğünüz gibi sadece program sonlandırmak için kullanmak zorunda değilsiniz.

Bunları “ek bilgi” olarak değerlendirdim, çünkü bu komutları (killall, pkill) süreç sonlandırmak için kullanmak zorunda değilsiniz, ama bizim başlığımız süreç sonlandırma.

Şimdi sonlandırma kısmına gelecek olursak; gördüğünüz gibi yanıt alamadığımız süreçleri uçbirim üzerinden sonlandırabiliyoruz. Hatta uçbirim üzerinden sonlandırmak için de tek alternatifimiz bu komutlar değil. Sanırım çoğumuz gerekli durumlarda sistemin anlık durumunu gözetlemek için “top” komutunu kullanıyoruzdur. Uçbirimden “top” komutu ile sistemi izlerken “k” tuşuna basın (burada da aynı durum mevcut, yani k kısayolu kill’den

geliyor fakat sadece sonlandırma işlemi yapmak zorunda değiliz). k tuşuna bastıktan sonra sizden bir pid numarası istenecek;

PID to kill

Buraya işlem yapacağınız sürecin pid numarasını girin (istenilen süreç listede olmayabilir de, pid numarasını öğrenmenin alternatif bir yolu için, ipuçları bölümüne bakabilirsiniz). Daha sonra sizden bir girdi daha istenecek;

Kill PID 3460 with signal [15]

Bu bölümde de bize sorulan sürece göndermek istediğimiz sinyalin numarasıdır. Yine 9 göndererek sonlandırabiliriz (SIGKILL).

Bir başka alternatif olarak htop komutu verip (sisteminizde htop yüklü değilse şu komut ile kurabilirsiniz; sudo apt-get install htop), işlem yapacağımız süreç üzerine gelerek “k” tuşuna bastığınızda size biraz daha görsel ve açık olarak göndermek istediğiniz sinyal sorulacak. Buradan istediğiniz sinyali seçebilirsiniz ayrıca gönderebileceğiniz diğer sinyal alternatiflerini de görebilirsiniz. Sonlandırma işlemi için SIGKILL sinyalinin en garanti olacağını belirterek geçelim.

Uzun uzadıya anlattığımız durumun özeti şu şekilde; bir süreci sonlandırmak istediğimizde yanıt alamazsak uçbirimden süreci sonlandırabiliriz. Uçbirimden süreci sonlandırabilmek için aynı kapağa çıkan çeşitli alternatiflerimiz var. pkill, killall, top, htop... Hangisi kolayımıza giderse onu kullanabiliriz.

Senaryo biraz daha kötüleşsin; uçbirime ulaşamıyor olalım. Yani masaüstümüz yanıt vermiyor, doğrudan X’in yanıt vermediği bir durumda kaldığımızı düşünelim. Bu durumda normal yollardan (menüden ya da alt + F2 ile) uçbirime ulaşmamız mümkün olmayacaktır. Aslında bu adım sadece bir noktada yukarıdaki işlemten fark gösterecektir.

Böyle bir durumda ihtiyaç duyacağımız araç; sistemi “kilitleyen” süreçleri sonlandırmak için kullanacağımız bir uçbirim. Yani eğer bir uçbirim bulabilirsek yukarıda bahsettiğimiz adımları uygulayabiliriz. Bu noktada yardımımıza koşacak olan araç “tty”. İzleyeceğimiz yol ise “ctrl + alt + F1” kombinasyonunu kullanarak tty1’e düşmek. Bu işlemden sonra bize giriş yapacağımız kullanıcı adı/şifre sorulacak. İşlemleri devam ettireceğimiz kullanıcı adını girerek uçbirime düşebiliriz. Yani artık bir uçbirime sahip oluyoruz. Sonlandırmak istediğimiz süreci istediğimiz/kolayımıza gelen herhangi bir yol ile sonlandırabiliriz.

Burada ek bilgi olarak şunu vermek de yararlı olacaktır sanırım; uçbirime düşmek için de tty1 tek alternatif değil. ctrl + alt + F(1, 2, 3, 4, 5, 6) bu şekilde F6’ya kadar olmak koşuluyla tty1, tty2...tty6 üzerinde işlem yapabilirsiniz hatta hepsine teker teker düşerek, hepsi üzerinde farklı kullanıcı ile işlem yapabilir ve aralarında geçiş yapabilirsiniz. Tekrar masaüstünüze dönmek içinse ctrl + alt + F7 kombinasyonunu kullanmanız yeterli olacaktır. Birden fazla masaüstü açık ise F8, F9.. şeklinde sırası ile geçişlerinizi yapabilirsiniz.

Durumu biraz daha ilerletip senaryoyu vahimleştirelim ve tty araçlarını da kullanmadığımızı varsayalım. Yine ihtiyaç duyacağımız araç bir uçbirim olacak. Fakat bulunduğumuz bilgisayar üzerinden işlem yapamıyor olduğumuz için bu sefer yardımımıza SSH geliyor. Bu gibi bir durumda ise SSH ile bilgisayarınıza bağlanarak, işlem yapmak istediğiniz bilgisayar üzerinde bir uçbirime sahip olabilirsiniz. Uçbirimi elde edebildikten sonra gerisi zaten birinci senaryo ile aynı duruma geliyor.

SSH kullanmayı bilmeyip, merak eden arkadaşlar SUDO’nun 12. sayısına göz atabilirler.

Bütün konuyu özetleyecek olursak, başımız sıkışınca yapmamız gereken bir uçbirime sahip olabilmek. Uçbirime menüden ulaşabiliriz. Eğer ulaşamazsak tty ile işimizi halledebiliriz. Bu araçlara da ulaşamıyorsak SSH ile sıkıntı yaşadığımız bilgisayara bağlanarak bir uçbirim elde edebiliriz.

Kolay gelsin herkese.