

Vim

Semetey Coşkun

Nisan, 2016

Vim'i hiç bilmeyen arkadaşlar da forumda sıkça verilen komutlardan birisi olarak görüyordur. Bazılarımız biliyordur fakat zor geliyordur, bazılarımız da kullanıyordur elbette. Ben kullananlar ile kullanmayanlar arasında kalıyorum tam olarak.

Vim, konsol tabanlı yani terminal üzerinde çalışan bir metin editörüdür. Daha sonraki zamanlarda kullanıcı arayüzü birimine entegre edilerek fare vs. gibi grafik arabirim kütüphanesi bileşenleri ile de uyumlu çalışabilecek hâle getirilmiştir. Hatta daha da ötesi Eclipse gibi ileri seviye ortamlara dahi eklentiler yazılarak buralarda da çalışabilecek hâle getirilmiştir.

Vim, bu yazımızın konusu olmakla beraber genelde sistemle ya da geliştirme ile uğraşanların kullandığı bir editörüdür. Aslında sadece metin editörü demeye içim el vermiyor açıkçası, çünkü bildiğimiz metin editörlerinden değildir kendisi. Örneğin kendi yazacağımız betiklerle kendisine özellikler ekleyebilmekte ve ayar dosyaları sayesinde çok fazla özelleştirebilmekteyiz.

Giriş kısmından nasıl bir aracı tanıtacağımız anlaşılmıştır. Pekiştirmek için şunun altını çizeyim, ben de konsoldan yani terminalden iş yapacağım zaman Vim'i tercih ediyorum, fakat sadece metin editörü olarak kullanıyorum. Tüm özelliklerini kullanmak nasip olmaz bir kişiye sanıyorum ama birçok özelliğini aktif olarak kullanmadığım için kendimi Vim kullanıcısı olarak göremiyorum ve bu nedenle kendimi kullananlar ve kullanmayanlar arasında sayıyorum.

Vim ile ilgili okuyabileceğiniz tüm yazılarda Vim ve diğer Vi klonlarının tarihçesini, hiyerarşisini ve ilişkilerini bulabilirsiniz. Bu nedenle bu aşamada sadece şunu söylemek ile yetineceğim; Vi, zamanında Unix sistemler için yazılmış olan bir metin editörü ve çok tutulduğu için bir klonu oluşturulmuş kendisinden sonra. Vim de aynı diğerleri gibi Vim'den klonlanarak ortaya çıkmış ve Vi uyumlu olarak, Vim'e özellikler eklene eklene son hâlini almıştır ve ismini de "Geliştirilmiş Vi" anlamına gelen "Vi IMproved" ibaresinden almıştır.

Gelelim konumuza, Vim...

Linux'a ilk başladığım zamanlar ve sonrasındaki uzun bir süre terminalden bir şeyler düzeltmem gerektiği zaman pico/nano gibi düzenleyicileri kullanıyordum. Bunlar biraz daha anlamlı geliyordu bana çünkü Vim'i anlamıyordum. Arayüz ile kullanılan editörler (Gedit gibi) fare ile başlık ve araç çubuğundaki menü ve butonlarla ile kontrol edilebiliyor. nano/pico de benzeri şekilde, fare ile kontrol edilemeyeceği için tuş kombinasyonları ile kontrol ediliyor. Benim aklıma yatan buydu. Fare olmadığına göre tuş kombinasyonları olmalı. Vim de durum tam olarak böyle değildi, ilk denemelerimde kapatamadım Vim'i. Çünkü Vim diğerlerinden farklı olarak "mod" gibi bir özellik barındırıyor. Örneğin yanlışlıkla yazma moduna girdiğiniz zaman Vim'den çıkması gerçekten kader kısmet işine dönüyor bilmeyen bir kişi için. Ki ben de bunlardan birisiyim. Birisiydim diyelim...

Fikir sahibi olmayan arkadaşların yazı takibini kaybetmemesi için hemen açıklayalım; "mod dediğimiz nedir": Vim ile bir metin dosyasını açtığınız zaman isteğe/ihtiyaca göre birden fazla modda bulunmak mümkündür. Özellikle eklentiler ile beraber mod sayısı epey fazla oluyor fakat biz özellikle ilk yazıda Vim Guru olmak niyetinde olmadığımız için temel modlardan üç tanesini ele alacağız.

Öncelikle normal mod ile başlayalım. Normal mod, standart olarak vim komutu ile bir metin dosyasını açtığımız zaman bizi karşılayan moddur.

```
1 vim örnek_dosya
```

komutu ile dosyamızı açalım, içi dolu bir metin dosyanız olduğunu ya da oluşturabileceğinizi varsayıyorum bu işlem için. örnek_dosya yerine istediğiniz bir metin dosyasını açabilirsiniz.

Şu an baktığınız ekran normal mod. "Standart bir terminal metin editörü arayüzü, normal mod dediğimiz nedir?" diye düşünenler varsa; örneğin rakamlardan birkaç tane rakam beğenerek yazmaya çalışabilirler ve güzelce başarisiz olabilirler. Normal modda istediğiniz gibi yazı yazamazsınız, ancak özel komut tuşlarına denk gelebilerseniz, bu durumda yazabilirsiniz; a tuşu, o tuşu, i tuşu gibi özel tuşlara basmadığınız sürece normal bir metin editörüne yazar gibi yazamazsınız Vim editörü ile.

Yanlış bir intiba uyanmasını istemem, normal modda iken Vim editöründe yazı yazılmaz, metin düzenlenmez, gibi bir intiba oluşması yanlış olur, çünkü normal modda iken "normal bir editör ile yazı yazar gibi yazamazsınız" fakat yazı yazabilir ve metni düzenleyebilirsiniz. Buna ileride değineceğim.

Bir diğerk modumuz ise komut modu. : (iki nokta üst üste) karakterine bastığımız zaman geçtiğimiz mod'dur. Vim editörünün kendine ait komutları vardır. Kendisine bu komutları vermek için öncelikle komut moduna geçmemiz gerekir. Sadece komut verme değil, metin araması yapma, bir metni farklı bir metin ile değiştirme vs. gibi işlemleri komut modu ile yapabiliriz. İki nokta üst üste karakterine bastıktan sonra istediğiniz gibi yazı girebildiğinizi görebileceksiniz tek bir farkla; yazıları imlecin bulunduğu yere değil sadece iki nokta üst üste işaretinden sonra girebiliyorsunuz (ekranın en altından bahsediyorum). Çünkü dosyayı düzenlemiyorsunuz sadece, Vim editörüne komut giriyorsunuz. Bu nedenle yazdığınız yazıları ekranın en altında iki nokta üst üste karakterinden sonra görüyorsunuz. "Vim'e hangi komutları verebiliyoruz?" sorusunu da ileriye atıyorum. Öncelikle modlarımızı bitirelim ve genel yapıyı anlamaya çalışalım.

Son modumuz olan (Vim editörünün değil elbette, bizim inceleyeceğimiz son mod) girdi modu. Yani Vim'i diğerk metin editörleri gibi kullanabileceğimiz mod, klavyeyi şimdiye kadar kullanmayı bildiğimiz gibi kullanabileceğimiz mod.

vim örnek_dosya komutunu verdikten sonra istediğimiz gibi komut giremiyorduk çünkü normal modda bulunuyorduk. Fakat yanlışlıkla a, o ve i tuşlarına bastıktan sonra girebileceğimizi söylemiştim yukarıda. Bu tuşlara bastıktan sonra yazı girebiliyoruz çünkü bu üç karakter de normal mod komutu olarak bir şekilde Vim editörünün girdi moduna geçmesini sağlıyor ve girdi modu klavye ile istediğimizi yazmamıza müsaade ediyor.

Az çok kafamızda oturdu sanıyorum ama toparlamak adına ufak bir özet geçmek istiyorum. Vim editör konusunu işliyoruz, bunda sorun yoktur sanırım. =) Vim, normal metin editörlerinden daha fazla özelliğe sahiptir ve bu özelliklerini sağlamak için kullanılan bazı modları vardır. Bunlardan en çok haşır neşir olacağımız normal mod, komut modu ve girdi modudur.

Bu yazımızı giriş niteliğinde kabul edeceğim ve tüm modlarda kullanabileceğimiz, yazıyı okuduktan sonra en azından sistem dosyalarını (ya da istediğimiz dosyaları) artık Vim ile düzenleyebilecek duruma gelmenizi planlıyorum.

Normal Mod ile devam edebiliriz. Bu mod bizi ilk karşılayan moddu. Bu modda iken ok tuşları ile ya da h, j, k, l tuşları ile yazı içerisinde karakter karakter gezinebilirsiniz (ok tuşları klavyenin bir diğerk tarafında durduğu için hızı kestiği söylenir genelde). Daha doğrusu bir tık desek daha anlamlı olur. Yukarı ok tuşuna basınca kavramsal olarak bir karakter ilerlemiş olmuyoruz çünkü fakat demek istediğimi anladınız sanıyorum. Gedit (Normal kullanıcı ara yüzü bir metin editörüdür.) ile gezerken fare yardımı ile ya da Page Down ve Page Up tuşları sayesinde binlerce satırlık dosyalar rahatça geziniyoruz, komut satırında bunu nasıl yapacağız? Vim kadar tutulan bir uygulamada elbette bu düşünülmüş hatta çok daha fazlası düşünülmüş. Ubuntu üzerinde iseniz ve Vim'i depodan kurduysanız (sudo apt-get install vim) Page Down ve Page Up tuşlarını ve fareyi kaydırma (scroll) işlemleri için kullanabilirsiniz. Öncelikle bunu belirteyim. Tabii Vim gibi bir uygulama kullanırken bir süre sonra fareyi unutacağımız için pek de gerek kalmayacak.

Karakter karakter ilerlemek dışında w harfi ile (sanıyorum word'un w'si) ve b harfi ile (bu da back'in b'si olmalı) de metin içerisinde gezinebilirsiniz. Bu gezinme ise normal bir metin editöründe ctrl + sağ/sol ok yönleri ile geziniyor gibi düşünebilirsiniz, kelime kelime ilerleme normal modda iken bu iki harf ile yapılmaktadır. e harfi ile de mevcut kelimenin sonuna gidebiliyoruz. Burada şuna dikkat edin lütfen, yazma yani girdi modunda değiliz, bastığımız karakterler dosyaya yazılmıyor, şu an bastığımız karakterler normal modda olduğumuz için bir Vim komutu ya da kısayolu olarak düşünülebilir.

Hâlâ normal moddayız ve dosyamızda geziniyoruz ve imlecin bulunduğu satırın sonuna gitmek istiyoruz; bu işlem için \$ işaretini kullanıyoruz (Alt Gr + 4 kombinasyonu). Bir önceki paragrafta hatırlattığım gibi sadece ilgili kombinasyona basarak imlecin hareketini izleyin, ekrana herhangi bir karakter yazmayacağız.

Farz edelim 1000000 satırlık bir dosyada 23415. satıra gitmek istiyoruz normal modda iken. Bu durumda kaydırma vs. gibi işlemlerle vakit kaybetmiyoruz elbette. 23415gg yazıyoruz ekrana ya da 23415G yazıyoruz. Bu iki komut aynı işlemi yapıyor ve bizi istediğimiz satıra götürüyor. Yukarıdaki hatırlatmayı tekrar yapacağım, normal moddayız ve yazdıklarımız ekranda çıkmıyor, sadece Vim tarafından yorumlanıyor ve kendisine uygun bir komut görürse girilen tuşlar arasında, bu komutu yerine getiriyor. gg ya da G ise (goto'nun G'si ile alakalı olduğunu düşündüğüm) satıra git anlamına geliyor. 5G dediğimiz zaman örneğin normal modda iken Vim bunu "5. satıra git" olarak anlıyor.

G komutunu tek başına ya da 1G kullanırsak da Vim tarafından sırası ile “dosya sonuna git” ve “dosya başına git” olarak yorumlanmaktadır.

H, M, L komutları da sırası ile “ekranın başına git”, “ekranın ortasına git” ve “ekranın sonuna git” şeklinde yorumlanmaktadır.

Az evvel bahsettiğimiz b ve w komutları da gg komutunda olduğu gibi parametre almaktadır. Bu komutlar (w ve b) kelime kelime hareket etmek için kullanılmaktaydı. 4w komutu “4 kelime ileri git” anlamına geldiği gibi 3b komutu da benzer şekilde “3 kelime geri git” anlamına gelir.

0 (sıfır) komutunun da imlecin bulunduğu satır başına gittiğini söyledikten sonra normal modda gezinme işlemlerini şimdilik sonlandırabiliriz diye düşünüyorum. Bu kadar gezmek yeterli olmalı. =)

Başlangıç seviyesi için, öncelikle dosyada rahatça gezinebilmemiz ve dosyayı düzenleyebilmemiz yeterli. Her modda bütün işlemleri yapabilmek için, biraz uygulamayı tanımak için ve kendimiz için zaman ayırmamız ve pratik yapmamız gerekecektir. Bu nedenle düzenleme işlemlerini girdi moduna bırakacağım.

Normal modda epey bir gezindik, bir dosyada bildiğimiz her yere gidebilecek durumdayız denebilir. Aradığımız yere geldik ve şimdi düzenleyeceğiz. Düzenleme işlemleri için girdi modunu kullanacağız. Bu kısımda şimdilik söyleyeceğim tek şey normal moddan girdi moduna nasıl geçeceğimizi bildirmek olacak. Normal moddayken i harfine basınca (Vim büyük küçük harf duyarlıdır arkadaşlar, fakat girdi yani orijinal dilinde “insert mode” a - geçmek için I ya da i harfine basmanız da geçerlidir.) Vim girdi moduna geçecektir ki zaten ekranın en altında – INSERT – ibaresini görebilirsiniz. Aşağıda bu ibareyi gördüğünüz anda anlayabilirsiniz ki artık girdi modundasınız. Klavyeyi bildiğiniz gibi kullanabilir, yazabilir, silebilir, düzenleme işlemlerini yapabilirsiniz.

Tekrar ediyoruz; “normal modda iken i komutu ile girdi moduna geçiyoruz”. Şu anda ne yazarsak yani klavyeden ne girersek Vim bunu ekrana yani imlecin bulunduğu pozisyona basıyor. Bu durumda soru şu; “Tekrar normal moda nasıl geleceğiz?” cevap da “Esc” tuşu ile olacak. Esc tuşuna bastığınızda en aşağıdaki – INSERT – ibaresinin kalkmış olduğunu görmelisiniz. İlginç alışkanlıktır genelde girdi modundan normal moda geçmek için Esc tuşuna bir kere basılmaz, garanti olsun diye sanırım seri bir şekilde Esc tuşu yoklanır, INSERT ibaresinin yok olduğundan emin olana kadar basılır. =)

Artık normal moddayız. Aslında hâlâ çok büyük soru işaretlerimiz bulunuyor. Dosyada istediğimiz yerlere gidip, girdi moduna geçip düzenlemeleri yapabiliyoruz. Şu an yapamadığımız yani aklımızda soru işaretleri ise “Girdi modunda klavyeyi bildiğimiz gibi kullandık, düzenledik fakat kaydedemedik?” bir diğer güzel soru ise “Hadi kaydettik diyelim, Vim’den nasıl çıkıyoruz?”

Son olarak kafamıza takılan sorular için komut modunu kullanacağız. Sırası ile bir dosyada normal modda ilerleyerek istediğimiz pozisyona geldik ve (normal modda iken i tuşuna basıp) girdi moduna geçerek düzenlemeleri yaptık. Daha sonra tekrar (Esc tuşuna basarak) normal moda geçtik ve girdi modunda iken yaptığımız değişikliklerin kaydedilmesini istiyoruz.

Modları tanıtırken komut moduna nasıl geçeceğimizden bahsetmiştik. Normal moddayken : karakterine yani iki nokta üst üste işaretine basıyoruz ve ekranın en alt satırında iki nokta üst üste işaretini görüyoruz. Burada komut moduna geldiğimizi anlıyoruz. Girdi moduna geçtiğimizde aynı yerde – INSERT – yazıyordu ve girdi moduna geçtiğimizi anlıyorduk.

Şu an ihtiyacımız olan şey dosyayı kapatmak ve yaptıklarımızı kaydetmek. Yani görsel bir metin editöründe çarpı tuşuna bastığımız zaman bize soruları yerine getirmek istiyoruz “Kaydet”, “Kaydetme”, “Vazgeç” gibi düşünebiliriz.

Komut modunda komutlarımızı yazıp “Enter” tuşuna basıyoruz ve Vim, “Enter” tuşundan sonra komutu gönderdiğimizi anlıyor.

Bir dosyada değişiklik yaptıktan sonra yaptığımız değişiklikleri kaydetmek için w (tahmin edileceği üzere write’ın w’si) yazıp hemen “enter” tuşuna basıyoruz ve yaptığımız tüm değişiklikleri kaydetmiş oluyoruz. q yazıp “enter” tuşuna bastığımızda ise (quit) dosyadan çıkış işlemi yapılıyor ve Vim’den çıkmış oluyoruz.

Yapılan değişiklikler daha kaydedilmemiş olan bir dosyadan çıkarken şunun kararını vermek gerekiyor “Değişiklikler kaydedilip mi çıkılacak” yoksa “Değişiklikler kaydedilmeden mi çıkılacak”. Çünkü kaydedilmemiş değişiklikler

olan dosyadan salt q komutu ile çıkamıyoruz. Ya kaydedip ya da kaydetmeden çık komutunu açıkça vermemiz gerekiyor.

“Kaydet çık” demek için kaydetme ve çıkma komutlarını komut modunda iken kombine ediyoruz yani wq yazarak (write and quit) çıkıyoruz. Tersî şekilde kaydetmeden çık diyeceksek eğer q! Komutunu girerek enter tuşuna basıyoruz.

Bu bilgiler ile dosyada gezinip, deęişiklik yapıp, deęişiklikleri kaydederek ya da kaydetmeden çıkabilmemiz gerekiyor. Elbette vim’in özellikleri ve yapabileceklerimiz bunlarla sınırlı deęil. Bu nedenle sonraki sayıda da Vim konusuna devam edeceğiz. O zamana kadar bu bilgilerle işinizi görebilmenizi umuyorum, en azından başlangıç olabileceğini umuyorum.

Herkese kolay gelsin.