Vim - Seri II

Semetey Coşkun

Nisan, 2016

Bu yazı geçen sayıda başlattığımız Vim serisinin devamı niteliğindedir ve geçen yazıyı okuduğunuzu ve artık vim ile bir dosyayı açabildiğinizi, düzenleyebildiğinizi ve en önemlisi açtığınız dosyaları kapatabildiğinizi varsayacağım. Elbette gerekli yerlerde hatırlatmaları ihmal etmem, o kadar da katı değilim bu konuda.

Geçen yazıda bahsetmiştik, birçok mod olmasına rağmen biz 3 tane mod üzerinde duracaktık.

1. Vim ile bir metin editörünü ilk açtığmız zaman bizi karşılayan Normal Mod'du
2. Normal modda iken "i" tuşuna bastığımızda geçtiğimiz ve ESC tuşu ile tekrar normal moda geçtiğimiz Düzenleme (orjinali insert) Modu'ydu. Bu modda iken klavyeyi bidiğimiz gibi kullanabiliyorduk yani klavyeden girdiğimiz tuşlar açtığımız metin üzerinde etkili oluyor, metin dosyası içinde yazı yazabiliyor ve silebiliyorduk.
3. Yazı serisinde üzerinde duracağımız ve ipuçları vereceğimiz diğer modumuz ise Komut Modu'ydu yani vim'e komutlar gönderdiğimiz ve istediğimizi komutlar sayesinde yapmasını sağlayabildiğimiz moddu.

Geçen yazıda daha çok metin dosyaları içerisinde gezinme üzerinde durmuştuk ve normal modda bahsettiğimiz komutların birçoğu metin dosyasında gezinme üzerineydi. Normal modda iken de metin dosyalarını düzenleyebildiğimizi fakat bundan ileride bahsedeceğimizi söylemiştim. İşte o gün bugün. Bugün düzenleme moduna hiç gitmeden, normal modda yani metin dosyasına girer girmez bizi karşılayan modda iken nasıl metin düzenleyebileceğimizden bahsedeceğiz ve ufak tefek bazı komut modu ipuçlarından bahsedeceğiz.

vim'in daima diğer metin editörlerinden daha üstün olduğu söylenir ve el alışkanlığı edinince (edinilebilince denmesi daha doğru bence)görsel metin dosyalarından da çok daha hızlıca istenilen işler halledilebilir denir. Ben buna katılıyorum fakat el alışkanlığı edinmenin ciddi bir pratik sonrasında olacağına inanıyorum. Tabi artık tüm uygulamalar eklenti eklenerek genişletilebiliyor ve istenilen özelliklerin eklenmesi sağlanabiliyor. Bu nedenle her bir metin editörü üzerinde uzman olmadığım için bu kadar iddialı konuşmayacağım fakat açık olmak gerekirse şahsi görüşüm bu söylenenin doğru olduğu yönünde.

Metin düzenleyici ile yapabileceğimiz işlemler zaten sınırlıdır, eğer bir IDE, bir tanımlayıcı vs amacı ile kullanmıyorsak. Bir metin dosyası içerisinde ya bir metin eklemesi yaparus yani yeni metinler yazabiliriz, ya damevcut metinler üzerinde düzenleme yapabiliriz, silebiliriz yani özetle.

```
1 vim örnek_dosya
```

komutumuzu veriyoruz ve üzerinde işlemler yapacağımız yani içerisinde metinler olan (önemli bir metin dosyası olmazsa iyi olur, silme işlemlerini yanlışlıkla da olsa kaydetme ihtimaline karşı). Bir metin dosyasını kaç türlü silebiliriz, aklıma gelenleri sıralayayım:

1. İleri doğru birer birer silebilirim bir metin dosyasını (normalde delete tuşu ile yaptığımız işlem)
2. 1'inci adımdaki aynı işlemi geriye doğru yapabilirim (normalde backspace denilen tuş ile yaptığımız silme işlemi)
3. Bir kelimeyi fare ile önce tararım, sonra taradığım kelimeyi silebilirim.
4. Bir satırı olduğu gibi tararım ve tüm satırı olduğu gibi metin dosyasından uçurabilirim.
5. Uçbirim kullananların pek sık kullandığı bir kombinasyon olduğunu tahmin ettiğim: imlecin bulunduğu yerden itibaren satır sonuna kadar olduğu gibi silme işlemi yapmak isteyebilirim.
6. 5'inci adımın zıttını yani imlecin bulunduğu bir pozisyondan itibaren olduğu gibi satır başına kadar silmek isteebilirim.

Bunlar aslında fazla bile fakat ihtiyaç duyulan özellikler gerçekten kod yazarken ya da bir ayar dosyasını düzenlerken. Bunlardan bazılarını gerçekten artık görsel metin editörleri de sağlıyor. Pek bilinmiyor ya da kullanılmıyor olsa da örneğin 3. ve 4. adımları normal silme işleminizi yaparken ctrl tuşuna basarak Gedit ile de yapabilirsiniz. Ama bizim konumuz vim olduğu için biz konumuza tekrar dönelim:

2'inci işlemi girizgahın istisnası olarak kabul edeceğim, düzenleme moduna geçmeden, normal modda düzenleme ibaresine yönelik bir istisna. Çünkü bu işlemi yapmak için komut moduna geçip bazı komutları vermemiz gerekiyor. Daha kolay olduğu için önce i tuşu ile düzenleme moduna geçmenizi ve akabinde istediğiniz ve bildiğiniz gibi backspace tuşunuzu kullanmanızı önereceğim.

1'inci adımdaki işlemi yapmak için normal modda iken "x" tuşunu kullanıyoruz. Düzenleme modunda iken klavyeyi zaten bildiğimiz şekilde kullandığımız için bu mooda iken ekstra bir bilgiye ihtiyacımız yok. Bunu artık zaten bildiğinizi kabul ederek daha sonraki normal mod komutlarında ilgili işlemi düzenleme modunda zaten normal şekilde yapabileceğiniz vurgusundan kaçınacağım.

Önceki yazımızdan hatırlayacağınız b,w komutları vardı. Metin içerisinde karakter karakter değil, sözcük sözcük ilerlemek için kullanmıştık bu iki kısayolu. Bu harfleri silme işlemleri için de kullanacağız, tek bir farkla, başlarına delete ibaresinin d harfini getirerek. b komutu bir normal modda gezinme esnasında önceki sözcüğe geçmemizi, w ise bir sonraki sözcüğe geçmemizi sağlıyordu. Bu durumda bir imlecin bulunduğu pozisyondan itibaren bir önceki sözcüğü silmek için db mevcut sözcüğü silmek için ise dw kısayollarını kullanacağız.

Olduğu gibi mevcut satırı silmek isteyecek olursak da dd kısayolunu kullancağız ve imlecin bulunduğu mevcut satırı olduğu gibi sileceğiz. Ya da imlecin bulunduğu mevcut satırda, olduğu satırı silmek yerine, mevcut imleç pozisyonundan itibaren sadece satır sonuna kadar olan bölümü silmek istersek d$ kısayolunu kullabiliyoruz.

Fark ettiyseniz, birinci yazıda bahsettiğimiz gezinme kısayollarının başına d koyduğumuz zaman, ilgili gezinme kısayolu silme kısayoluna dönüşüveriyor.

Geçen sayıda belirtmediğimiz bir gezinme kısayolundan daha bahsetmiş olalım öncelikle; $ kısayolu ile imleci bulunduğu satırın sonuna konuşlandırabiliyorduk. Bunun tam tersi için ^ kısayolunu kullanabiliyoruz. Yani imlecin bulunduğu mevcut satırın, başına gitmek için ^ kısayolunu kullanmamız yeterli. Tahmin edeceğiniz gibi d^ kısayolu ise, bir önceki silme ipucumuzun tersi şekilde, imlecin bulunduğu satır için, imleciğin bulunduğu poziyondan itibaren, satır başına kadar olan bölümü silmek için kullanılıyor.

Burada silme komutlarını verirken dikkat etmemiz gereken önemli bir husus var: dd komutu. dd komutu seri bir şekilde verilebiliyor. Yani d tuşunu basılı tuttuğunuzda, vim bunları seri bir dd komutu olarak algılıyor ve sıra sıra satırları silmeye başlıyor. Dikkat etmemiz gereken nokta ise şu oluyor; d den sonra bir karakter girilmesi gereken silme kısayollarında biraz seri olmak gerekiyor, aksi halde d'ye bastıktan sonra uzun süre beklenecek olursa bunlar dd komutu olarak algılanacaktır.

İlgili değişikliklerimizi yaptıktan sonra vim'den çıkmak için komutlarımız vardı; w, q, wq ya da q! olarak belirtmiştik bunları bir önceki yazıda. Tabi çıkmak zorunda değildik, sadece yapılan değişikliklerin saklandığından emin olmak için salt w komutunu da verebiliyorduk. Bunlar komut modunda verilen komutlardı. Yukarıda metin dosyasını tamamen normal modda düzenledik, bu nedenle çıkış için de yine normal mod kısayollarını kullanmak isteyebiliriz. Bu komutların açıkmalarını vermeyeceğim, sadece geçen yazıda üzerinde durduğumuz, komut modu muadillerini vereceğim.

```
1 ZQ = :q!
2 ZZ = :wq
```

Metin dosyalarında gezinme üzerinde durmuştuk önceki yazıda, bu yazıda ise normal modda iken bir dosyayı düzenleme ve işlemleri kaydederek ya da kaydetmeden çıkış yapma üzerine durduk. Metin dosyaları ile ana işlemler arasında tek eksiğimiz "arama işlemleri" kaldı.

Normal modda ya da düzenleme modunda iken doğrudan / tuşuna basarak, akabinde aramak istediğiniz metini girebilirsiniz. Fakat vim büyük küçük harf duyarlıdır, bu nedenle örneğin "Configuration" şeklinde bir metin parçası ararken "configuration" yazarsanız aradığınızı bulamazsınız, bu durumda iki seçeneğiniz var, ya istediğinizi tam olarak doğru şekilde bilecek ve yazacaksınız ya da daha güzel ve modern bir alternatif olarak, harf duyarlılığını kapatacaksınız. Vim editöründe harf duyarlılığını komut moduna geçerek kapatabiliyoruz ya da açabiliyoruz. Hatırlarsınız normal moddan komut moduna geçmek için : tuşuna basmamız gerekiyordu. Gerçi düzenleme modunda : tuşuna basarsanız gerçekten metin içerisine : yazmış olursunuz. Yani düzenleme modunda iseniz önce ESC tuşu ile normal moda geçip akabinde : tuşuna basarak komut moduna geçiniz ve:

```
1 :set ignorecase
```

komutu ya da kısaltılmışı olan:

```
1 :set ic
```

komutunu aramalarda büyük harf küçük harf duyarlılığını kapatmak için kullanabiliyoruz (komutu yazıp enter tuşuna basınız, buradaki : komutun bir parçası değil, komut modunda yazmanız gerektiğine dikkat çekmek için kullanılan bir işarettir). Aramalardaki bu durumun iptali için ise:

```
1 :set noic
```

komutunu kullanmamız yeterli. Bu komuttan sonra aramalarımız tekrardan büyük küçük harf duyarlı hale gelecektir.

Her mod ile ipucu vermeye özen gösterdiğimiz için komut modunda da, metin dosyası içerisinde gezinmekten bahsetmeden geçmeyelim. vim tabi ki komut modunda da gezinebilmeyi mümkün kılıyor. Bunun için komut modunda iken herhangi bir rakam yazmanız yeterli. Örneğin bir metin dosyası içerisinde hangi satıra gitmek istediğinizi biliyorsanız eğer o rakamı yazarak gidebilirsiniz, örneğin 1986. satıra gitmek istiyorsanız, önce normal modda iken : tuşuna basarak komut moduna geçtikten sonra ilgili rakamı yazıp, girdi (enter) tuşuna basmanız yeterli:

```
1 :1986
```

Bir adım ilerisi olarak, bir metin dosyasında gitmek istediğiniz karakteri biliyorsanız eğer, yukarıdaki örnek için, örneğin metin dosyasının 1986. karakterine gitmek istiyorsanız, komut moduna geçtikten sonra;

```
1 :goto 1986
```

Komutunu vermeniz yeterli. Bunlara ek olarak bu gezinmeler esnasında, dosyada satır sayılarının da görünmesini istiyorsanız eğer, "set number" ya da kısaltılmışı olan "set nu" komutunu kullanabilirsiniz.

```
1 :set number
```

Komutunu verdikten sonra satır numaralarını, satır başlarında görebilirsiniz. Bu komutun iptali için ise;

```
1 :set nu!
```

komutunu vermeniz yeterli.

Komut modu ile ilgili de gezinme ipuçlarını verdikten sonra tekrar arama işlemlerine dönecek olursak, büyük-/küçük harf duyarlı ve duyarsız arama yapmaktan bahsetmiştik son olarak. Bu yazının son ipucu olarak arama sonuçlarında gezinme üzerinde duracağız. Örneğin "configuration" anahtar kelimesini metin içerisinde aradım ve 10 tane "Configuration" bulundu diyelim ya da bulunacak farz edelim. Arama işlemini gerçekleştirdikten sonra yani normal mod ya da komut modunda iken

```
1 /configuration
```

yazıp, girdi tuşuna bastıktan hemen sonra, imleç ilk "Configuration" bulunan yere konuşlandırıldı diyelim. Bu aşamadan sonra, tekrar arama işlemi yapmamak için, bir sonraki "Configuration" ibaresi geçen yere imleci konuşlandırmak için n (next'in n'si) kısayolunu kullanıyoruz. Böylece bulunan ibareler arasında gezinebiliyoruz. n tuşuna basa basa 6. Configuration ibaresi bulunan yere geldim diyelim ve 5. pozisonya tekrar gitmek istersek de p kısayolunu kullanıyoruz (bu da previous'un p'si sanıyorum).

Son ipucu olarak, normal modda arama ile komut modunda arama arasındaki fark olarak, normal modda;

```
1 /configuration
```

yazıp ararsanız imleç doğrudan aranan kelimeye konuşlanır. Komut modunda arasanız da;

```
1 :/configuration
```

imleç doğrudan aranan kelimenin bulunduğu satır başına konuşlanır.

Bir sonraki yazıda genel olarak bahsettimizi toparladıktan sonra seviyeyi arttırarak seriye devam edeceğiz.

Şimdilik herkese kolay gelsin diyerek bırakıyoruz.