

Editörler: Kedit vs. ve En Önemlisi: vi

04

- **vi: Visual Editor**
 - **vi Bir Kelime İşlemci Değildir!**
 - **Beğenseniz de Beğenmeseniz de, vi Öğrenmelisiniz!**
 - **Dosya Açmak / Yaratmak**
 - **vi Editörünün “Durum”ları**
 - **gvim**

LINUX işletim sistemiyle birlikte kullanılacak pek çok editör vardır. Bunlar KDE, GNOME gibi masa üstü yöneticileri altında kullanılacak grafik arabirimli editörler ve grafik olmayan ekranlarda kullanılacak konsol editörleridir.

KDE ile beraber yüklenen dosya editörlerini

KDE Menüsü

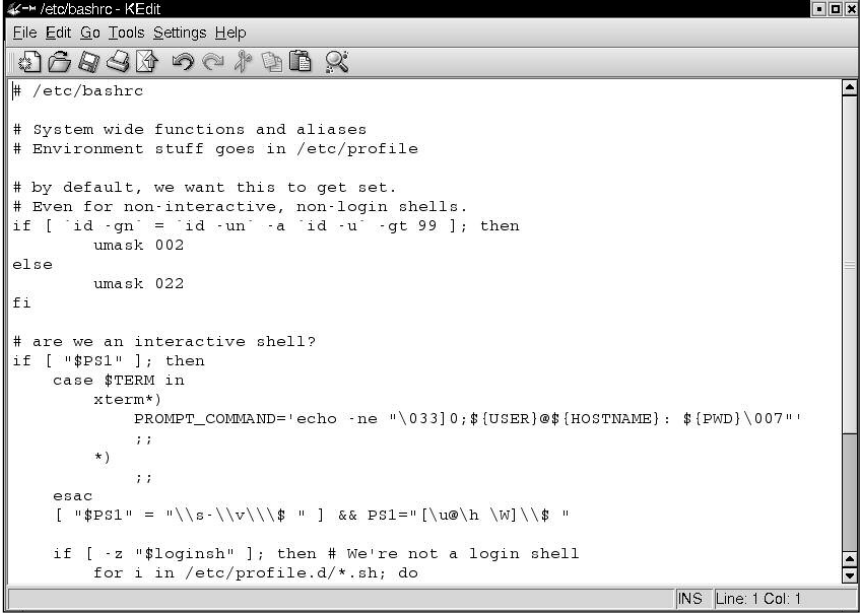
– More Applications

– Editors

altında bulabilirsiniz.

Kim Korkar LINUX'tan?

Bunları deneyip zevkinize uygun birini seçebilirsiniz. Bu menüdeki grafik kullanıcı arabirimi (GUI) editörlerin kullanımı son derece kolaydır.



```
# /etc/bashrc

# System wide functions and aliases
# Environment stuff goes in /etc/profile

# by default, we want this to get set.
# Even for non-interactive, non-login shells.
if [ `id -gn` = `id -un` -a `id -u` -gt 99 ]; then
    umask 002
else
    umask 022
fi

# are we an interactive shell?
if [ "$PS1" ]; then
    case $TERM in
        xterm*)
            PROMPT_COMMAND='echo -ne "\033]0;${USER}@${HOSTNAME}: ${PWD}\007"'
            ;;
        *)
            ;;
    esac
    [ "$PS1" = "\s-\v\\\$ " ] && PS1="\u@\h \w\\\$ "

    if [ -z "$loginsh" ]; then # We're not a login shell
        for i in /etc/profile.d/*.sh; do
```

Konsol editörlerine gelince durum biraz değişir. Bunlar grafik ekranlar daha bilgisayar dünyasında ortaya çıkmadan yazılmış editörler olup her türlü bilgisayar terminalinde çalışacak şekilde geliştirilmiştir. Şimdi diyeceksiniz ki “grafik ekranı olmayan bilgisayar mı kaldı ki?” Kaldı tabii.. Kalmaya da devam edecek gibi görünüyor. Niye bilgisayar denince yalnızca PC aklınıza geliyor? Hiç ekranı olmayan bilgisayarlar bile var. Bu nedenle konsol editörleri hala profesyonel bilgisayarlıların; daha doğrusu UNIX’cilerin eli ayağıdır.

Konsol editörleri arasında en çok sevilenler **pico**, **emacs** ve **vi** editörleridir. Bunlar arasında tüm UNIX uyarlamalarında standart olanı **vi** editörüdür. Bu nedenle UNIX dünyasına adımını atan herkesin **vi** öğrenmesi oldukça önemlidir.

vi: Visual Editor

vi Bir Kelime İşlemci Değildir!

vi bir editördür! Yalnızca basit metin dosyaları üzerinde kullanıldığında işe yarar. Font, kalın yazı, italik yazı gibi kavramlar **vi**'de yoktur. Yani mektuplarınızı, hele tekliflerinizi ve raporlarınızı **vi** ile yazmaya çalışmak anlamsız olacaktır. Bu gibi işler için OpenOffice, KOffice gibi yazılımlar daha uygundur.

Eğer kelime işlemci gereksiniminiz bilimsel makalelere yönelik ve özellikle matematiksel sembollerle doluyorsa o zaman TeX, LaTeX, Lyx üçlüsünü öğrenmenizi öneririz. Düzleme çizilebilecek ve yazılabilecek her türlü dökümanı uluslararası standartlarda geliştirebilirsiniz.

Beğenseniz de Beğenmeseniz de, vi Öğrenmelisiniz!

vi UNIX dünyasının standart editörüdür. UNIX veya herhangi bir türeviyle çalışan her bilgisayarda bulunur.

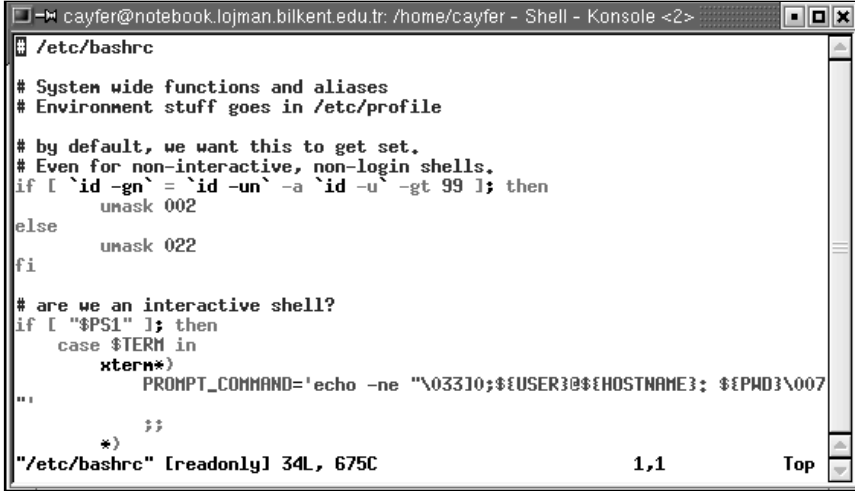
vi ile ilk karşılaşan kullanıcılar genellikle “*bu ne ya..*” diye tepki gösteriyorlar. Bunlar arasında sabırlı olanlar bir süre sonra **vi** kullanmayanlara hayretle bakmaya başlıyorlar. Çok güçlü bir editör olmasına rağmen bu kitapta **vi** hakkındaki herşeyi anlatmayacağız. Zaten pek anlatamayız da.. **vi** için ayrıca bir kitap yazmak gerekir. Önce temel editör işlemlerinden başlayalım isterseniz...

Dosya Açmak / Yaratmak

Bir dosyayı düzenlemek üzere açmak için

vi dosya-adı

komutunu verebilirsiniz. Örneğin “**vi /etc/bashrc**” komutunu verdiğinizde karşınıza



```
caayfer@notebook.lojman.bilkent.edu.tr: /home/caayfer - Shell - Konsole <2>
# /etc/bashrc
# System wide functions and aliases
# Environment stuff goes in /etc/profile

# by default, we want this to get set.
# Even for non-interactive, non-login shells.
if [ `id -gn` = `id -un` -a `id -u` -gt 99 ]; then
    unask 002
else
    unask 022
fi

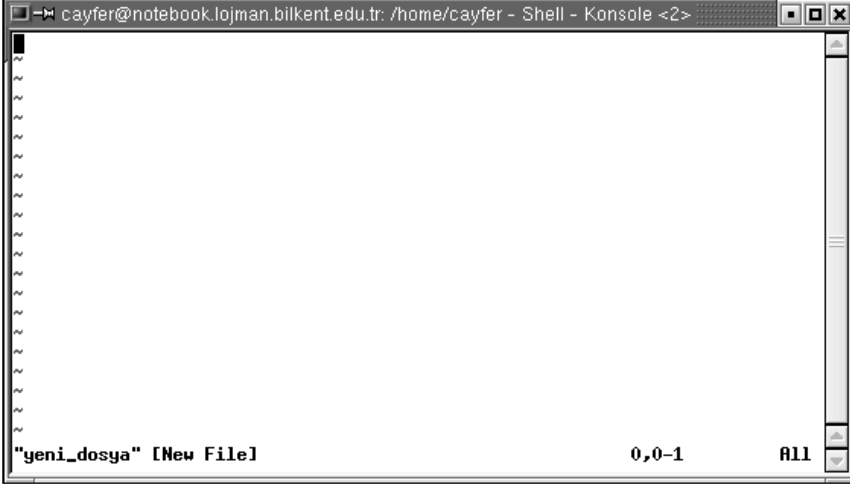
# are we an interactive shell?
if [ "$PS1" ]; then
    case $TERM in
        xterm*)
            PROMPT_COMMAND='echo -ne "\033I0;${USER}@${HOSTNAME}: ${PWD}\007
            ;;
        *)
            ;;
    esac
fi

"/etc/bashrc" [readonly] 34L, 675C          1,1          Top
```

gibi bir ekran gelecektir.

Ekranın sol alt tarafında düzenlenen (edit edilen) dosyanın adı, yalnızca okuma için ([readonly]) açıldığı, satır sayısı ve kaç byte uzunluğunda olduğu belirecektir. Bu örnekte **/etc/bashrc** dosyasının yalnızca okuma için açılmasının nedeni “**root**” kullanıcı dışındakilerin bu dosya üzerinde yazma yetkisi olmamasındandır.

Eğer **vi** komutuna parametre olarak verdiğiniz dosya diskte yoksa, yeni ve boş bir dosya yaratılarak karşınıza



ekranı gelecektir. Bu ekrandaki ~ işaretleri, boş satırlar ile olmayan satırlar karışmasını diye görüntülenir.

vi Editörünün “Durum”ları

vi editörü herhangi bir anda iki durumdan birindedir:

1. Ekleme durumu
2. Komut durumu

“Ekleme Durumu”nda, klavyeden bastığınız herşey imlecin bulunduğu noktadan başlayarak dosyaya eklenecektir.

“Komut Durumu”nda klavyeden bastığınız herşey bir **vi** komutu olarak değerlendirilecektir.

vi editörünü ekleme durumuna sokmak için “**i**” tuşuna (**küçük i**: *insert*) bir kez basmanız yeterlidir.

Ekleme durumundan komut durumuna geçmek için ise Esc tuşuna bir kez basmanız gerekmektedir. Esc tuşuna birden fazla kez basmanız bir şey değiştirmeyecektir. Yani komut durumundayken Esc tuşuna basarsanız gene komut durumunda kalırsınız.

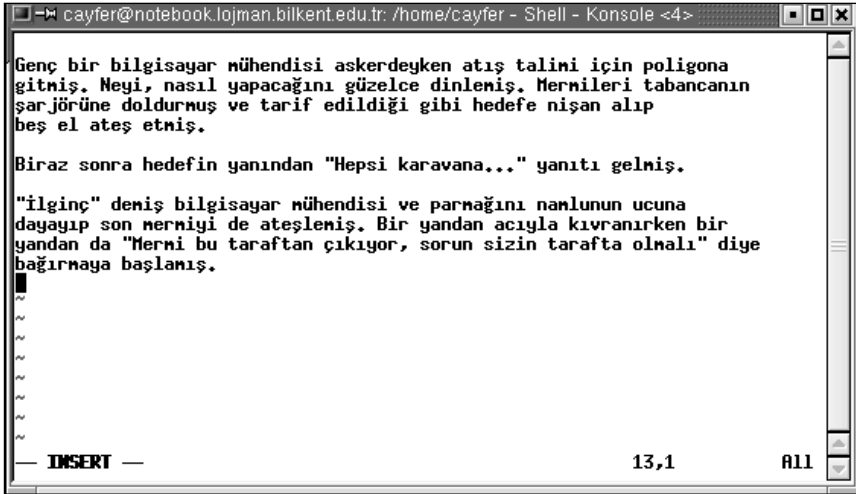
Kim Korkar LINUX'tan?

vi ekleme durumundayken (*insert mode*) ekranın sol alt tarafında “-- **INSERT** --” işaretini görürsünüz.

Sağ tarafta ise dosyanın kaçınıcı satırında ve o satırın hangi yatay pozisyonunda olduğunuzu izleyebilirsiniz.

Ekleme Durumu

Dosyanız ekleme durumundayken (bu duruma geçmek için bir kez “**i**” tuşuna basmalısınız) birkaç satır veri giriniz. Satırların sonuna geldiğinizde Enter tuşuyla satır başı yapabilirsiniz.



```
cayfer@notebook.lojman.bilkent.edu.tr: /home/cayfer - Shell - Konsolle <4>
Genç bir bilgisayar mühendisi askerdeyken atış talini için poligona
gitmiş. Neyi, nasıl yapacağını güzelce dinleniş. Mermileri tabancanın
şarjörüne doldurmuş ve tarif edildiği gibi hedefe nişan alıp
beş el ateş etmiş.

Biraz sonra hedefin yanından "Hepsi karavana..." yanıtı gelmiş.

"İlginç" demiş bilgisayar mühendisi ve parmağını namlunun ucuna
dayayıp son mermiyi de ateşleniş. Bir yandan acıyla kıvrılırken bir
yandan da "Mermi bu taraftan çıkıyor, sorun sizin tarafta olmalı" diye
bağırılmaya başlanış.

```

Komut Durumu

Dosyayı kaydetmek için yani editöre “**kaydet**” (*write*) komutunu verebilmek için ekleme durumundan (hatırlarsanız en başta bir küçük **i** harfi ile bu duruma geçmiştiniz) çıkıp komut durumuna geçmeniz gerekecektir. Ekleme durumundan çıkmak için bir kez Esc tuşuna basmanız yeterlidir. (Fazla Esc basmanın bir zararı olmaz. Sisteminiz her fazla basış için bir kez düdük çalacaktır; o kadar.)

Ekranda komut durumunda bulunduğunuza ilişkin bir işaret aramayın. Yoktur! Ekleme durumunda olmamanız komut durumunda olduğunuzu gösterir. Ancak, hangi durumda bulunduğunuzdan emin olmak istiyorsanız, bir kez Esc tuşuna basıp komut durumunda olmayı garantileyebilirsiniz.

Kim Korkar LINUX'tan?

de değildir. Eğer dizin belirtmediyseniz o dosya çalışma dizininizde de olmayabilir. Bir başka olasılık da, dosya adını yazarken büyük harf-küçük harf konusuna dikkat etmemiş olabilirsiniz; hatırlarsanız LINUX işletim sisteminde **deneme** ile **Deneme** farklı dosya adlarıdır!

Aşağıdaki mesajlardan birini görürseniz:

[Read Only] veya
File is Read only veya
Permission denied

söz konusu dosyaya ya da bulunduğu dizine yazma yetkiniz yok demektir. Normal olarak yapabileceğiniz bir şey olmadığından **vi** seansını bir an önce durdurup, dosyanın sahibi ile görüşmeniz ya da uygun bir kullanıcı kimliğine bürünmeniz gerekmektedir.

File System full mesajını görürseniz, çalıştığınız disk ya da kotanız dolmuş demektir; gereksiz dosyaları silerek yer açmanız gerekmektedir.



UNIX işletim sisteminde, kullanıcıların disklerde kullanabilecekleri alanların toplam büyüklüğünü sınırlama (kota koyma) olanağı vardır. Böyle bir sınırlamanın olup olmayacağına, olacaksa her kullanıcı için kotanın kaç megabyte olacağına sistem yöneticisi karar verir.

Kullanıcılar kotalarını doldurduklarında bir mesajla uyarılır ve genellikle kotalarını biraz aşmalarına izin verilir; ancak birkaç gün içinde tekrar kota limitlerinin altına inmezlerse, sistem daha fazla yeni dosya yaratmalarına veya eski dosyalarını büyütmelerine izin vermeyecektir.

vi editörü, terminal penceresinin tamamını kullanan **tam-ekran** (*full screen*) bir editördür. Bu nedenle temel işlevler için en iyi hakim olunması gereken komutlar, imleci ekranda dolaştırma komutlarıdır.

İmleç Dolaştırma Komutları

Bu komutların verilmesi sırasında, editör komut konumunda bulunmalıdır. (Yani editör yeni başlatılmış olmalı veya ekleme konumundan çıkmak için Esc tuşuna basılmış olmalıdır. Hatırlayacaksınız, fazla Esc basmanın bir zararı yoktu; dolayısıyla bulunduğunuz konumla ilgili bir şüphemiz varsa, komut durumuna geçmek için hiç çekinmeden Esc tuşuna basınız.)

Normal koşullarda imleci ekranda dolaştırmak için klavyedeki ok tuşlarını kullanabilirsiniz. Ama UNIX dünyasında ne zaman nasıl bir terminal veya terminal emülasyon programı kullanacağınız belli olmaz; bu nedenle ok tuşları bulunmayan bir terminali kullanırken gerekebilecek imleç dolaştırma komutlarından söz etmekte yarar var:

vi İmleç Yönlendirme Komutları	
Tuş	İmleç Hareket Yönü
Oklar	...
h	Sola bir karakter.
j	Aşağı bir satır.
k	Yukarı bir satır.
l	Sağa bir karakter. (Küçük L harfi)

Tabii bu arada Return (veya Enter) tuşu ile BackSpace tuşunun da sırasıyla satırbaşı ve bir geri anlamına geldiğini hatırlatmakta fayda var.

Birçok **vi** komutu gibi, imleç hareket komutlarının da başına bir sayı koyarak, bu sayı kadar sağa, sola, aşağı ve yukarı hareket sağlanabilir.

3h	3 karakter sola.
2k	2 satır yukarı, gibi.

Birerli adımlar dışında hareket sağlayan ve çok kullanılan imleç dolaştırma komutlarından bazılarıysa:

o	(Sıfır) İmlecin bulunduğu satırın başına.
ş	İmlecin bulunduğu satırın sonuna.
w	Bir sonraki sözcüğün başına.
b	Sözcüğün başına. (Ya da bir önceki sözcüğün başına)

Ekleme Komutları

Editör kullanırken en çok yapılan işlerden biri, eski bir metnin başına, sonuna ve araya satır/kelime/harf eklemektir. **vi** editöründe ekleme konumuna geçmenin yöntemlerinden birini (**i** komutu) daha önce belirtmiştik. Yine de tekrarlamak istiyoruz:

i	İmlecin, üzerinde bulunduğu karakterin hemen solundan başlayarak, Esc tuşuna basıncaya kadar basılan her karakteri metne ekler. Varsa, eski metin sağa doğru itelenir.
----------	--

Diğer ekleme komutları:

a	İmlecin, üzerinde bulunduğu karakterin hemen sağından başlayarak, Esc tuşuna basıncaya kadar, basılan her karakteri metne ekler. Varsa, eski metin sağa doğru itelenir.
A	İmlecin, üzerinde bulunduğu satırın sonundan başlayarak, Esc tuşuna basıncaya kadar, basılan her karakteri metne ekler.



Bu komutların büyük veya küçük harflerle verilmesi farklı ve önemlidir. Dikkatli olun.

Yazı Silmek

Daha önce yazılmış metin parçalarını silmek için kullanılan komutları aşağıdaki tabloda örnekleriyle özetlenmiş olarak bulacaksınız:

Metin Silme Komutları	
x	İmlecin üzerinde bulunduğu tek karakteri siler.
3x	İmlecin üzerinde bulunduğu karakter dahil, sağa doğru 3 karakter siler.
dw	İmlecin bulunduğu yerden kelime sonuna kadar siler. Eğer imleç sözcüğün başındaysa, sözcüğü siler.

2d̄w	İmlecin bulunduğu yerden başlayarak 2 sözcük siler.
dd	İmlecin bulunduğu satırı siler.
2dd	İmlecin bulunduğu satır dahil, aşağı doğru iki satır siler.
D	İmlecin bulunduğu yerden satır sonuna kadar siler. (d̄s komutuna eşdeğerdir)
d̄s	Satırın sonuna kadar siler. (D komutuna eşdeğerdir)

Yanlışlıkla Silerseniz...

Eğer hatanızı hemen fark ederseniz, **u** komutuyla (*undo*) son silme işlemini geri döndürebilirsiniz. Undo aslında yalnızca son silme işlemini değil, son değişikliği geri döndürür.

Undo komutunu **U** harfiyle (büyük U) verdiğinizde en son değişiklik yapılmış olan satırdaki tüm değişikliklerin hepsi birden geri alınır.

Metin Bloklarının Yerini Değiştirmek

vi editörüyle metin bloklarının yerini değiştirmek istediğinizde kullanacağınız yöntem kes-yapıştır yöntemidir. Yerini değiştireceğiniz metin bloğunu önce bulunduğu yerden, uygun bir komutla (örneğin tek satır için **dd** komutu gibi) silmeli (kesme işlemi), daha sonra **p** komutuyla (*put*) yeni yerine yapıştırmalısınız. Eğer yapmak istediğiniz işlem bir metin bloğunu silmekse, kesme işleminden sonra başka bir yere yapıştırmamanız yeterlidir. Bu işlemleri bir örnekle anlatmak daha kolay olacak:

Aşağıdaki örnek metindeki, “- **Rüyalarınız**” sözcüğüyle başlayan üçüncü satırı en üst satıra taşımak istediğinizi varsayalım. Önce imleci bu satırın üzerine getirmelisiniz. (**dd** komutuna hazırlık.)

Kopyalama ve yer deęiřtirmede kullanılan **p** (küçük p) yapıřtırmayı imlecin bulunduęu satırın altına, **P** (büyük p) ise üstüne yapar.



Son Komutu Tekrarlamak

Bir nedenle, son deęiřiklik komutunuzu başka yerlerde de tekrarlamamız gerekirse, “.” (nokta) komutu ile bunu yapabilirsiniz.

Metin Eklemenin / Deęiřtirmenin Birkaç Deęiřik Yolu

Komut durumundayken basılacak tuřlarla bazı önemli metin deęiřtirme iřlevleri yerine getirilebilir. Bunlar arasında sıkça kullanılanları ařaęıdaki tabloda bulabilirsiniz.

Komutların hangi İngilizce sözcüklerin kısaltmaları olduęunu da İngilizce bilen okurların hatırlamalarını kolaylařtırmak için tabloya ekledik.

Bazı Önemli Metin Deęiřtirme Komutları		
Komut	İngilizcesi	Açıklama
I	Insert	(Büyük I) İmlecin bulunduęu satırın BAŐINA eklemeye başla. (Esc’e kadar)
o	Open Line	(Küçük o) İmlecin bulunduęu satırın ALTINA bir boş satır aç ve oraya eklemeye başla.
O	Open above	(Büyük O) İmlecin bulunduęu satırın ÜSTÜNE bir boş satır aç ve eklemeye başla.
s	Substitute Char	İmlecin bulunduęu yerdeki karakteri sil ve yerine yeni metni eklemeye başla. (Esc’e basıncaya kadar)
S	Substitute Line	İmlecin bulunduęu satırını sil ve yerine yeni metni eklemeye başla.

r	Replace Char	İmlecin bulunduğu karakteri bir sonra basılacak karakterle değiştirir.
R	Replace Text	İmlecin bulunduğu noktadan itibaren, yeni metni eski metnin üzerine yerleştirir. (Esc'e basıncaya kadar)
J	Join	(Büyük j) İmlecin bulunduğu satırla arkasındaki satırı birleştirir.
cw	Change Word	İmlecin bulunduğu sözcüğü, yeni girilecek sözcükle değiştirir.



Birçok **vi** komutu, başına bir çarpan yerleştirilerek birden fazla kez tekrarlanabilir. (Aynı cebirdeki gibi...)

Örneğin, satır silmek için kullanılan **dd** komutu, **5dd** şeklinde verildiğinde "5 satır sil" anlamına gelir. Komutların bu özelliğini daha önceki örneklerde fark etmiş olmalıydınız.

Metnin İçinde Dolaşmak

Şu ana kadar hep birkaç satırdan oluşan dosyalarda çalıştık. Hayat her zaman bu kadar kolay değildir. Özellikle elektronik posta için metin yazarken ya da program geliştirirken, sık sık metin içinde sayfa sayfa (bir başka deyişle ekran ekran) ileri-geri gitmek gerekir. Bu işlemler için kullanılan komutları burada kısaca bir tablo halinde sıralayacağız. Birer kere denemenizi öneririz. Üzerinde çalışmak için uzun bir dosya hazırlamak yerine, sisteminizin **/etc** dizinindeki **termcap** dosyasını kendi dizininize kopyalayıp bu dosya üzerinde çalışabilirsiniz. ("**cp /etc/termcap ~**" sonra "**vi termcap**")

Metin İçinde Dolaşma Komutları		
Komut	İngilizcesi	Açıklama
Ctrl-F	Forward	Bir ekran ileri.
Ctrl-B	Backward	Bir ekran geri.
Ctrl-D	Down	Yarım ekran ileri.
Ctrl-U	Up	Yarım ekran geri.
Ctrl-R	Redraw	Ekranı yeniden düzenle. (Çalışırken sistemden gelebilecek mesajlar ekranınızdaki görüntüyü bozarsa bu komut, vi ekranını ekranı silip yeniden oluşturarak düzenleyecektir.)
Ctrl-Y		Ekranı bir satır aşağı kaydır, imleç yerinde kalsın.
Ctrl-E		Ekranı bir satır yukarı kaydır, imleç yerinde kalsın.
z RETURN		(Küçük z) İmlecin bulunduğu satır ekranın en üstüne gelecek şekilde ekranı düzenle.
z.		(Küçük z ve nokta) İmlecin bulunduğu satır ekranın ortasına gelecek şekilde ekranı düzenle.
z-		(Küçük z ve eksi) İmlecin bulunduğu satır ekranın en altına gelecek şekilde ekranı düzenle.
H	Home	Ekranın en üst satırına git.
M	Mid Screen	Ekranın orta satırına git.
L	Lower Screen	Ekranın en alt satırına git.
RETURN		Bir sonraki satırın ilk karakterine git.

Yukarıda açıklanan komutların bazı özel kullanımları vardır. Bunlar ender kullanılan özellikler olup, burada sadece bir fikir vermek amacıyla birkaç örnek vereceğiz:

200z RETURN 200. satırı ekranın en üstüne getir.
4H Ekranın en üst satırının 4 altındaki satıra git.
5L Ekranın en alt satırının 5 üstündeki satıra git.

Metnin İçinde Arayarak Dolaşmak

Diyelim ki, program yazıyorsunuz ve değişiklik yapmak istediğiniz satırın yeri hakkında bir fikriniz yok. Tek hatırladığınız, değiştirmek istediğiniz satırda '**kayit_sayısı++**' diye bir karakter dizisi var. Bu diziyi içeren bir satır bulmak için komut durumunda,

/kayit RETURN

yazmanız yeterli olacaktır. İmleç, içinde bu dizi geçen ilk satırda duracaktır. Arama, imlecin bulunduğu yerden ileriye doğru yapılır. Eğer geriye doğru arama yapmak isterseniz "/" yerine "?" karakterini kullanmanız gerekir.

Diyelim ki, programın ilk rastladığı **kayit** sözcüğü, sizin ilgilendiğiniz değil ve aramaya devam etmek istiyorsunuz. Bu durumda **n** (*next*) tuşuna basmanız yeterlidir. Tüm arama komutunu yeniden yazmanız gerekmez. Eğer aramanın yönünü değiştirmek isterseniz **n** yerine **N** komutunu veriniz. Eğer aramanın halen hangi yönde olduğunu hatırlamıyor, fakat aramayı ileriye doğru yönlendirmek istiyorsanız / tuşuna, tam tersi içinse ? tuşuna basabilirsiniz.

Bulup Değiştirmek

Diyelim ki, bulunduğunuz satırda "**Linux**" diye bir sözcük var ve bu sözcüğü "**LINUX**" olarak değiştirmek istiyorsunuz. Bu işi yapmak için imleci "**i**" harfinin üzerine götürerek **4rINUX** komutunu vermeniz yeterli olacaktır. Bu yöntemde bulma görevini siz; değiştirme görevini ise **vi** üstlenmiş oluyor. Her iki işi de **vi**'in yapmasını istiyorsanız, değişikliğin yapılmasını istediğiniz satırın üzerine gelip,

:s/Linux/LINUX veya
:s/inux/INUX

komutlarını verebilirsiniz. Bu komutlar yalnızca ilk rastladıkları "**Linux**" ka-

rakter dizisini “**LINUX**” dizisiyle ve “**linux**” dizisini “**INUX**” dizisiyle değiştirecektir.

Bir satırdaki tüm “**Linux**” karakter dizilerini “**LINUX**” olarak değiştirmek için,

:s/Linux/LINUX/g

komutunu kullanmalısınız.

Şimdi, **s** komutuna birkaç ileri düzey örnek vererek bu komutun başka yeteneklerini de göstermek istiyoruz:

İleri Düzey “s” Komutu Örnekleri	
Komut	Açıklama
:1,100s/Unix/UNIX/g	Dosyanın 1. ve 100. satırları arasında rastlanan tüm “ Unix ” dizilerini “ UNIX ” olarak değiştir.
:1,\$s/Ugur/Ugur Ayfer/g	Dosyanın 1. ve sonuncu satırları arasında rastlanan tüm “ Ugur ”ları “ Ugur Ayfer ” olarak değiştir.
:%s/Ugur/Ugur Ayfer/g	% işareti tüm dosya anlamındadır.
:%s/teh/the/gc	Dosyadaki tüm “ teh ”leri “ the ” olarak değiştirir; ancak her bir değişiklik için kullanıcı onayı ister. (c: <i>confirmation</i>)
:g/Ayfer/s/Ugur/U./g	Tüm dosyada “ Ayfer ” dizisini arar (g/Ayfer/); “ Ayfer ” bulunduğu her satırdaki tüm “ Ugur ” dizilerini “ U. ” ile değiştirir.
:%s/Perl/\U&/g	Dosyadaki tüm “ Perl ” dizilerini “ PERL ” ile değiştirir. (\U büyük harfe dönüştürür; & ise, aranan diziyi baştan yazmamak için kullanılan bir kısaltmadır.)

<code>:g/^\$/d</code>	<p>Tüm boş satırları bulur ve siler. (^ işaretini satır başı anlamına; \$ ise satır sonu anlamına gelir. Bu örnekte içinde hiçbir karakter olmayan (başı ve sonu yan yana olan) satırlar silinecektir; boşluk karakterleri içeren satırlar bu kalıba uymayacağından silinmeyecektir. Eğer içinde boşluk karakterleri olan satırları da silmek istiyorsanız, <code>:g/^ */d</code> komutunu kullanmalısınız. (^ işaretinden sonraki boşluğa dikkat!)</p>
<code>:g/^/mo</code>	<p>Çılgın bir örnek!</p> <p>Bir dosyadaki satırların sırasını ters çevirir... (Son satırı birinci satır, sondan ikinci satırı ikinci satır,...). Nasıl mı? Başlangıcı olan her satırı (^) (zaten her satırın bir başlangıcı vardır) sıfırinci satıra taşır. (mo, move anlamındadır) Bu kısacık komutun bu işi yapacağına inanmıyorsanız bir deneyin.</p>

Metnin İçinde Satır Numaralarını Kullanarak Dolaşmak

vi editörü, normalde ekranda satır numaralarını göstermez. Eğer metnin içinde satır numaralarını kullanarak dolaşmanız gerekiyorsa (40. satıra git, sonra 75. satıra git, vs.), ekranda satır numaralarını görmek çok yararlı olacaktır.

Satır numaralarını ekranda görebilmek için:

:set number

Sıra numarasını bildiğiniz bir satıra gitmek için kullanacağınız komut:

nnG (nn numaralı satıra git, örneğin **55G**, **134G**) olacaktır.

Kim Korkar LINUX'tan?

Eğer sık kullandığınız **vi** komutlarına ilişkin bir kısaltma tanımlamak istiyorsanız, bu tanımlamanızı **map** komutuyla yapmanız gerekir. Örneğin,

```
map ^Y dd
```

tanımlaması, Ctrl-Y tuşunun, bulunduğunuz satırın silinmesini sağlayan **dd** komutu ile eş anlamlı olarak kullanılmasını sağlar.

Eğer **map** komutu ile F2 fonksiyon tuşuna “tüm boş satırları silme” komutunu tanımlamak isterseniz,

```
map #2 :g/^$/d
```

komutunu kullanabilirsiniz. **map** ile yapılmış tanımlamaları iptal etmek için **unmap** komutunu kullanmalısınız. (**unmap #2** gibi)

Bu tip kısaltmalarınızın kalıcı olmasını istiyorsanız **.vimrc** dosyası ile ilgili bölümü okuyunuz.

vi Başlatırken Verebileceğiniz Komutlar

Çok önemli olmamakla birlikte, editörü başlatırken komut satırından verebileceğiniz birkaç **vi** komutu vardır.

```
vi +230 mektup1.mail      mektup1.mail dosyasını aç ve imleci  
230. satıra götür.
```

```
vi + telefonlar            telefonlar dosyasını aç ve imleci son  
satıra götür.
```

Dosya İşlemleriyle İlgili Komutlar

vi editörü ile yarattığınız veya üzerinde çalışarak değişiklikler yaptığınız dosyaları diske kaydetmek ve buna benzer işlemler için kullanılan komutlar aşağıda sıralanmıştır.

Bu komutları kullanmadan önce komut durumuna geçmiş olmanız gerekmektedir: Komut durumuna geçmek için en az bir kez Esc tuşuna basmanız gerektiğini hatırlıyorsunuz, değil mi?

Dosya İşlemleriyle İlgili Komutlar	
Komut	Açıklama
zz	Dosyayı son haliyle diske kaydet ve vi 'dan çık. (:wq komut dizisine eşdeğerdir.)
:q	Dosyada değişiklik yapılmayacak, vi 'dan çık. (<i>quit</i>)
:q!	Yapılan değişikliklerden vazgeçildi, dosyayı değiştirmeden vi 'dan çık. (<i>quit</i>)
:w	Dosyayı diske kaydet. (vi 'da kal) (<i>write</i>)
:wq	Dosyayı diske kaydet ve vi 'dan çık. (<i>write-quit</i>)
:x	vi 'dan çık, değişmişse dosyayı diske kaydet. (<i>exit</i>)
:w dosya2	Üzerinde çalışılmakta olan dosyayı, " dosya2 " adıyla diske kaydet. (<i>write</i>)
:1,100w bölüm1	Üzerinde çalışılmakta olan dosyanın ilk 100 satırını " bölüm1 " isimli bir dosyaya kaydet.
:r dosya3	" dosya3 " adlı dosyayı oku ve imlecin bulunduğu noktadan başlayarak araya ekle. (<i>read</i>)

vi İçinden UNIX Komutu Vermek

Bazen, **vi** programıyla bir dosya üzerinde çalışırken, geçici olarak kabuğunuza dönüp, başka bir UNIX komutu çalıştırmanız gerekebilir. Diyelim ki, üzerinde çalışmakta olduğunuz dosyanın içine, bir başka dosyayı kopyalamanız gerekti, ama bu dosyanın tam adını hatırlayamadınız. **ls** komutunu bir kullanabilerseniz, bu dosyanın adını hemen hatırlayacaksınız. Böyle durumlar için, **vi**, size kabuk programınıza bir çıkış olanağı vermektedir. Bu olanaktan yararlanabilmek için Esc tuşu ile komut durumuna geçip,

!ls

komutunu veriniz.

UNIX'in zerafetine bir örnek: Diyelim ki, bir program için kullanım kılavuzu yazıyorsunuz ve kılavuzunuzun bir bölümüne, söz konusu programın bir çıktısını eklemek istiyorsunuz.

Kullanabileceğiniz komut,

:r !prog

Bu komutu verdiğinizde, **prog** isimli program çalıştırılacak ve çıktısı **vi** ile düzenlenmekte olduğunuz dosyada, imlecin bulunduğu noktaya yerleştirilecektir. (Sanki diskten bir dosya okumuşsunuz gibi...)

Birden Fazla Dosyayı Peşpeşe İşlemek

vi programını başlatırken, dosya adı olarak birden fazla parametre verebilirsiniz. Örneğin,

vi dosya1 dosya2 veya

vi dosya*

Bu durumda, **vi** önce **dosya1** isimli dosyayı düzenlenmek üzere ekrana getirecektir. Bu dosyayla işiniz bitip de,

w komutuyla birinci dosyayı (**dosya1**) kaydettikten sonra (write)

n komutuyla ikinci dosyaya (**dosya2**) geçebilirsiniz. (next)

.vimrc Dosyası

.vimrc dosyası, vi programıyla ilgili özel tercihlerinizi belirttiğiniz dosyadır. Eğer kişisel dizininizde .vimrc isimli bir dosya (dosya adının başındaki noktaya dikkat ediniz) varsa, vi programını her başlattığınızda, bu dosyanın içindeki vi komutları otomatik olarak çalıştırılacak ve böylece tercihleriniz ayarlanmış olacaktır. .vimrc dosyası basit bir text dosyası olup, vi dahil her türlü editörle yaratılabilir. Bu dosyada yer alabilecek bazı vi komutlarına örnekler vermek istiyoruz.

<code>map ^Y dd</code>	Ctrl-Y “satır sil” anlamında.
<code>map #2 :g/^\$/d</code>	F2 tuşu, “boş satırları sil” anlamında.
<code>ab cua Can Ugur Ayfer</code>	“cua” bir kısaltma olarak kullanılmış.

Daha Ayrıntılı Bilgi İçin...

vi programı hakkında daha yazılabilecek çok şey var. Eğer ilginizi çekiyorsa, O'Reilly Media'nın *Learning the vi Editör* (yazarı Linda Lamb; ISBN 0-937175-67-6) isimli kitabını hararetle tavsiye ederiz.

vi Komutları Özeti			
Komut	Açıklama	Komut	Açıklama
A	Satır sonuna eklemeye başla	x	İmlecin bulunduğu karakteri sil
a	İmlecin sağına eklemeye başla	dd	İmlecin olduğu satırı sil
I	Satır başına eklemeye başla	d3	İmlecin olduğu yerden 3 karakter sil
i	İmlecin soluna eklemeye başla	d\$	İmleçten sonrasını sil
O	Bu satırın üstüne satır ekle	u	Son değişikliği iptal et (Undo)
o	Bu satırın altına satır ekle	U	Satır için Undo
b	Bir önceki kelimenin başına git	r	Tek karakter değiştir
w	Bir sonraki kelimenin başına git	cw	Kelime değiştir
e	Bir sonraki kelimenin sonuna git	cc	Satırın tamamını değiştir
\$	Satır sonuna git	^	Satır başına git
H	Ekranın başına git	L	Ekranın sonuna git
^F	Sayfa ileri (Ctrl-F)	^B	Sayfa geri (Ctrl-B)
S	Satırın tamamını değiştir	:w	Dosyayı kaydet (w yeni-isim de olabilir)
/dizi	Dosyada " dizi "yi bul (ileriye doğru)	:q	Çık (Quit)
?dizi	Dosyada " dizi "yi bul (geriye doğru)	zz	Kaydet ve çık
:nu	Satırları numarala	:q!	Değişikliklerden vazgeç ve çık
	Blok taşımak için: <ul style="list-style-type: none"> • Blok başına gidiniz. • Örneğin 8dd ile sekiz satır siliniz • Taşımaya gideceğiniz yere gidip, p komutunu veriniz. 		Blok kopyalamak için: <ul style="list-style-type: none"> • Blok başına gidiniz. • Örneğin 8yy ile tampon belleğe sekiz satır alınız • Kopyalayacağınız yere gidip, p komutunu veriniz.

Bu sayfanın fotokopisini çekinmeden çekebilirsiniz. Dava açmayız...

gvim


Eğer **vi**'in gücünden yararlanmak, ama grafik ortamda çalışıyor olmanın avantajlarını da kaybetmemek istiyorsanız, **gvim** isimli mükemmel bir editör daha var. Bu editör **vi**'in bütün özelliklerini korurken aynı zamanda fare kullanımını da destekleyerek görsel birçok yenilik getiriyor.

gvim, “Graphical vi Improved” (görsel ve geliştirilmiş vi) sözcüklerinden elde edilmiş bir kısaltmadır. Terminal ekranınızdan **gvim** yazarak çalıştırabileceğiniz gibi bu editörü KDE menüsünde Applications – Editors altında da bulabilirsiniz.

Bütün **vi** komutları **gvim**'de de aynen çalışır. Ek olarak farenizi kullanarak imlecin yerini değiştirebilir ve metin üzerinde değişiklikler yapabilirsiniz. Kes-yapıştır türü hareketler bu sayede daha rahat olur.

Eğer 3 tuşlu bir fareniz varsa X kullanırken pencereler içinde veya arasında metin kopyalamak çok kolaydır. Fareyle seçtiğiniz her metin parçası otomatik olarak kopyalanır ve farenizin orta butonunu tıkladığınızda imlecin bulunduğu yere yapıştırılır. Ayrıca Ctrl-C, Ctrl-V gerekmez. Eğer KDE kullanıyorsanız Klipper programı da kes-yapıştır işlemlerine bir sürü ek özellik ekler. Eğer üç tuşlu bir fareniz yoksa, genellikle iki tuşa birden basarak aynı sonuca ulaşabilirsiniz. Fare ayarlarındaki “emulate 3 buttons” seçeneği bunun içindir.



Ekranınızın sağ altındaki  ikonuna tıkladığınızda önceden kopyaladığınız 10 metin parçasına ulaşabilirsiniz. (Tabii ki bu sayıyı ve Klipper'in daha birçok özelliğini değiştirebilirsiniz.)

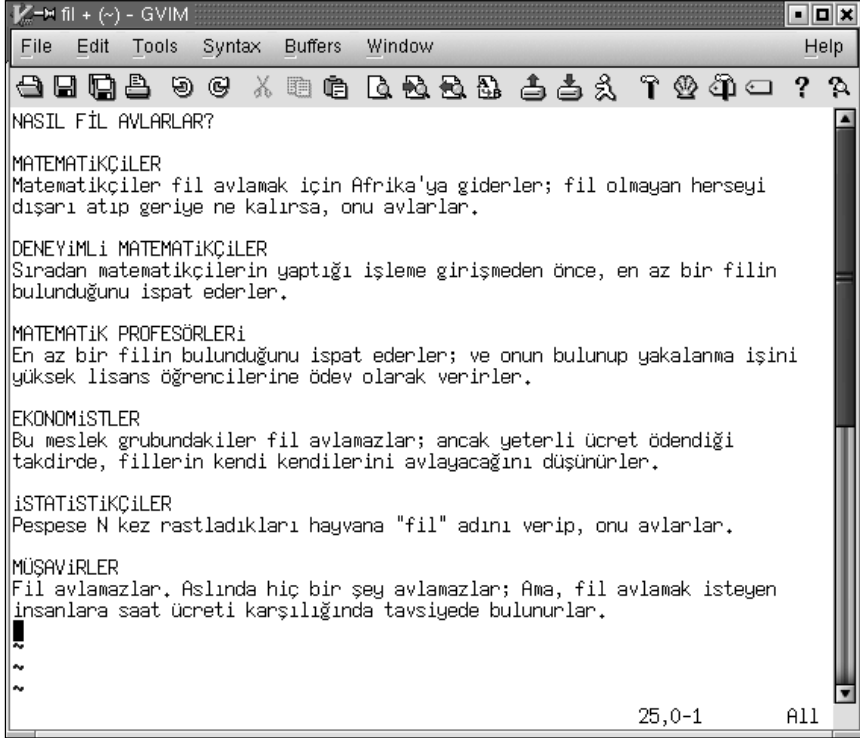
gvim'in en hoş özelliklerinden biri, aklınıza gelebilecek her türlü programlama diline uygun yazım kuralı seçilebilme özelliğidir. Eğer geleneksel dosya isimleri kullanıyorsanız **gvim** kullanarak açtığımız dosyalar zaten belirtilen dosya uzantısına uygun dilin yazım kurallarına göre düzenlemeler yapacak şekilde açılacaktır.

Örneğin adı “.c” diye biten bir dosyayı açtığınızda **gvim** dosyanızı otomatik olarak C dilinin yazım kurallarına göre renklendirecektir. Böylece çeşitli parantez, tırnak gibi açıldıysa kapatılması gereken gramer unsurları kolay al-

Kim Korkar LINUX'tan?

gılanabilecek şekilde ekranınıza gelecektir. **gvim** programının "Syntax" menüsünü kullanarak bu renklendirmenin hangi programlama diline göre yapılacağını seçebilirsiniz.

vi'dan farklı olarak **gvim** kullanırken karşınızda bir yardım menüsü vardır. **gvim** ile ilgili herşeyi bu yardım menüsünden ulaşacağınız dökümanların içinde bulabilirsiniz.



The screenshot shows a Gvim window titled "fil + (~) - GVIM". The menu bar includes "File", "Edit", "Tools", "Syntax", "Buffers", "Window", and "Help". The main text area displays a help menu for the word "fil" (file) in Turkish. The menu items and their descriptions are:

- NASIL FİL AVLARLAR?**
- MATEMATİKÇİLER**
Matematikçiler fil avlamak için Afrika'ya giderler; fil olmayan herşeyi dışarı atıp geriye ne kalırsa, onu avlarlar.
- DENEYİMLİ MATEMATİKÇİLER**
Sıradan matematikçilerin yaptığı işleme girişmeden önce, en az bir filin bulunduğunu ispat ederler.
- MATEMATİK PROFESÖRLERİ**
En az bir filin bulunduğunu ispat ederler; ve onun bulunup yakalanma işini yüksek lisans öğrencilerine ödev olarak verirler.
- EKONOMİSTLER**
Bu meslek grubundakiler fil avlamazlar; ancak yeterli ücret ödendiği takdirde, fillerin kendi kendilerini avlayacağını düşünürler.
- İSTATİSTİKÇİLER**
Pespese N kez rastladıkları hayvana "fil" adını verip, onu avlarlar.
- MÜŞAVİRLER**
Fil avlamazlar. Aslında hiç bir şey avlamazlar; Ama, fil avlamak isteyen insanlara saat ücreti karşılığında tavsiyede bulunurlar.

At the bottom right of the window, the text "25,0-1" and "All" is visible.