

# Tmux: Kuvvetli Uçbirim Yönetimi

Heartsmagic

Ağustos, 2011

# İçindekiler

1	Giriş . . . . .	2
2	Tmux ve Screen Nedir? . . . . .	3
2.1	Oturum Kurtarma . . . . .	3
2.2	Pencere Çoklama . . . . .	3
2.3	Pencereleri Yönetme . . . . .	3
2.4	Oturum Paylaşma . . . . .	4
3	Tmux ve Screen Arasındaki Farklar . . . . .	5
4	Tmux'un Kullanımı . . . . .	7
4.1	Oturum Yönetimi / Oturumdan Ayrılma, Geri Bağlanma / Oturum Kurtarma . . . . .	7
4.2	Pencere Çoklama / Yönetme . . . . .	9
4.3	Oturum Paylaşma . . . . .	13
4.4	Durum Çubuğunu Özelleştirme / Kopyalama,Yapıştırma Kipini Kullanma . . . . .	14

# 1 Giriş

Tmux bir Screen muadilidir, yani uçbirim çoklayıcısı. Bir uygulamayı başka bir uygulamanın muadili olarak tanıtmak demek örnek alınan, yani dengi olarak kabul edilen uygulamayı genel kullanımda belki de “de facto” olarak kabul etmek demektir. Özellikle de örnek yine özgür yazılım camiasının içinden bir uygulamaysa. Kısacası, yazımızın konusu olan Tmux’un (Y.N: Okuyabildiğim her şeyi Türkçe okumaya gayret gösteririm) örnek aldığı Screen saygı duyulacak ve hatta saygı duyulması gereken bir uygulamadır.

Screen uygulamasını daha önce kullanmayanlar için Tmux’u anlatmak zor olduğu kadar, bilenler içinse bir hayli kolaydır. Bu nedenle öncelikle Tmux ve Screen uygulamalarının ne işe yaradıklarını anlatarak yazımıza giriş yapalım. Screen’i kullanıyor olanlar veya bilenler doğrudan aralarındaki farkı anlatan bölüme geçebilirler.

## 2 Tmux ve Screen Nedir?

Bu uygulamalar uçbirim için birer pencere yöneticisi ve oturum çoklayıcısıdır. Cümle biraz havada kalıyor olabilir, bu nedenle gündelik örneklerden yola çıkarak uygulamaların biraz daha kafamızda yer etmesini sağlayalım.

### 2.1 Oturum Kurtarma

Uzak bir sunucumuz olsun ve bu sunucuyu SSH üzerinden yani uçbirimi kullanarak yönetiyor olalım. Yaptığımız işse bir yapılandırma dosyasını uzun uzadıya düzenlemek olsun. Öyle ki neredeyse bitirmek üzereyiz. Fakat öyle bir an gelsin ki artık yazamadığımızı fark etmiş olalım. Bizim veya sunucumuzun bağlantı sorunu nedeniyle oturumumuz sonlanmış ve biz bunca uğraşımızın üzerine yaptığımız değişiklikleri kaybetmiş olalım. Canımız sıkılır değil mi? Evet, tedbirli bir sunucu yöneticisi ya kullandığı düzenleyicinin kendiliğinden kaydetme özelliğini etkinleştirir ya da ara ara yaptığı bu değişiklikleri kendisi kaydeder. Peki, ikisini de yapmadığı zamanlar olmaz mı? Emin olun ki olur. İşte bu esnada bizim uygulamalarımız devreye girerek hayat kurtaran bir can simidi oluyorlar. Tmux ve Screen, uçbirim kapansa, bağlantımız kesilse bile arkada çalışmaya devam ederler. Kısacası tekrar bağlandığımızda bu uygulamaların oturumlarına geri dönmemiz ve dosyamızı kaldığımız yerden düzenlemeye devam etmemiz mümkündür.

### 2.2 Pencere Çoklama

İlk madde bu bölümün başında havada kalan cümleyi açıklamıyor gibi sanki. Gerçi oturum çoklama meselesinin biraz yanından geçiyor ancak yine de tam anlamıyla izah etmiyor, örneklemiyor. İlk maddeye bilerek oturum kurtarma daha doğrusu oturuma devam edebilme özelliğini koydum, zira çok zamanlar hayat kurtarıcı olacaktır bu özellik. Peki pencere çoklama ne demektir?

Aynı sunucumuza yine SSH ile bağlı olduğumuzu ve uçbirimden işimizi gördüğümüzü düşünelim. Öyle bir yapılandırma esnasındayız ki ikinci bir uçbirime bağlanma ihtiyacımız oldu bu sunucu üzerinde. Elimiz yerelde kullandığımız uçbirime gider, yeni bir SSH isteği yaparak sunucumuza bağlarız. Öyle mi? Bunu ancak taze veya bu ihtiyacını karşılamak için araştırma yapmayan tembel sunucu yöneticileri yapar :) Diğerleri ise elindeki çoklayıcı üzerinden bir pencere daha açar. Hem de bir iki tuş darbesiyle. Kısacası aynı SSH oturumunda, aynı uçbirimde (daha doğrusu Tmux veya Screen oturumunda) ikinci bir pencere oluşturur, işini buradan görmeye devam eder. Aynı zamanda bu iki pencere arasında da bir ileri bir geri gezinebilir. Güzel değil mi?

### 2.3 Pencereleri Yönetme

İkinci madde de bir hayli ilgi çekici görünmekte. Fakat son cümlelerdeki ileri geri gitme meselesi sanki azıcık zahmetli duruyor. Şöyle ifade edelim. Elimizde sürekli çıktı üreten bir penceremiz olsun. Bu pencerenin açık kalması gerekmektedir ve biz burada akan çıktıya göre bir yapılandırma dosyasını düzenlemeliyiz. Tamam işte, ikinci pencereyi açabiliyoruz ya, hemen yapalım. İkinci pencerede çıktıyı izlerken birincisinde dosyamızı açıyor ve düzenliyoruz. İkinci pencereye geçiş yapıp çıktıya bakıyoruz, birinci pencereye dönüyoruz, tam düzenlemeyi yapacağız sırada çıktıyı unutuyoruz :) Olmaz demeyin, sunucuda yoğun işlemler esnasında her şeyin olabileceği vakidir. Şimdi meselenin neden biraz zahmetli olduğu anlaşılıyordur.

İşte, yine tam bu durumda uygulamalarımız ihtiyacımızı karşılamak için imdadımıza yetişiyorlar: Pencereleri bölme, boyutlandırma, özelleştirme. Yeni pencere oluşturmak güzeldi, peki oluşturduğumuz bir pencereyi bölerek kullanabiliyor olmamız daha güzel olmaz mı? Zahmet çektiğimiz konuyu düşünelim. Penceremizin üst tarafında istediğimiz çıktı akıyor, bizse bölerek kullandığımız alt tarafta dosyamızı bu çıktıya göre düzenliyoruz. Hatta aynı pencerenin yan bölmesinde de "top" komutu ile sunucuyu gözlemliyoruz! Sanırım aşağıdaki resim ne demek istediğimi tam olarak ifade edecektir.

Resimden de görülebileceği üzere pencerelerimizi dilediğimiz gibi şekillendirmemiz mümkün. Sunucu üzerinde bu şekilde oluşturulmuş bir çalışma ortamı bize oldukça rahatlık sağlayacaktır.

```

serkan@one:~$
22ab sp bfd7eb60 error 4 in libunityshell.so[2f3a000+1c7000]
Aug 13 16:05:56 one gnome-session[1422]: WARNING: Application 'compiz.desktop'
killed by signal
Aug 13 16:05:56 one gnome-session[1422]: WARNING: App 'compiz.desktop' respaw
ing too quickly
Aug 13 16:05:56 one gnome-session[1422]: CRITICAL: We failed, but the fail wha
le is dead. Sorry...
Aug 13 16:06:01 one dbus[884]: [system] Activating service name='org.debian.ap
t' (using servicehelper)
Aug 13 16:06:01 one AptDaemon: INFO: Initializing daemon
Aug 13 16:06:01 one dbus[884]: [system] Successfully activated service 'org.de
bian.apt'
Aug 13 16:06:01 one AptDaemon: INFO: UpgradeSystem() was called with safe mode
: 1
Aug 13 16:06:01 one AptDaemon.Trans: INFO: Simulate was called
Aug 13 16:06:01 one AptDaemon.Worker: INFO: Simulating trans: /org/debian/apt/
transaction/62901f065f31420eab5a15b3b0b19cde
Aug 13 16:06:03 one AptDaemon.Worker: INFO: Upgrade system with safe mode: 1

# /etc/fstab: static file system information.
#
# Use 'blkid -o value -s UUID' to print the universally unique identifier
# for a device; this may be used with UUID= as a more robust way to name
# devices that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options> <dump> <pass>
proc /proc proc nodev,noexec,nosuid 0 0
# / was on /dev/sda5 during installation
UUID=fc4345d0-3e73-42b7-9c58-0433b94eae9a9 / ext4 errors=remou
nt-ro 0 1
# /media/Ambar was on /dev/sda6 during installation
UUID=2e3655e9-7ff2-4e46-93a8-04e918e6799c /media/Ambar ext4 defaults
0 2
# /media/natty was on /dev/sda2 during installation
#UUID=c20c35fb-4bc9-4e31-a19a-3619a3e0c518 /media/natty btrfs defaults
0 2
# swap was on /dev/sda3 during installation
UUID=0e48b0de-e452-4951-b0af-bd2edbe34def none swap sw
0 0

1,1 Tümü
one 0:vim*

```

Şekil 1:

## 2.4 Oturum Paylaşma

Bir arkadaşımızın sistemine girmemiz ve onun bizden rica ederek yapmamızı istediği şeyi halletmemiz gerekiyor. Kolay bir şekilde SSH ile bağlanıp, işimizi halledip çıkabiliriz. Peki arkadaşımız biz bu işi hallederken neler yaptığımızı görmek istiyorsa, hem de birebir olarak? Bu gibi bir durumda oturum paylaşma kullanılabilir. Aynı Tmux veya Screen oturumuna bir başkası bağlanıp, bizim yaptıklarımızı görebilir, oturum ortak olduğu için gerektiğinde kendisi de müdahale edebilir. Bu özelliği, verdiğimiz örnekteki şekliyle kullanabileceğimiz gibi, eğitim maksatlı da kullanabiliriz. Sunucu yönetenler için ortak çalışma şekli bile olabilir bu özellik. Bizim yaptıklarımız birebir karşı tarafa da yansıtılacağı için bir nevi sohbet aracı olarak bile kullanılabilir oturum paylaşma. Sadece "enter" yerine ctrl+c kullanmamız gerekir, onu da sistemi sürekli "Böyle bir komut bulunamadı" iletisini basma işkencesinden kurtarmak için yaparız :)

Görülebileceği üzere uçbirim üzerinde kullanılan pencere çoklayıcılarının güzel ve hayat kurtarıcı özellikleri mevcut. İki uygulama da temel anlamda birbirlerine benzemektedir. Zira yaptıkları iş hemen hemen aynı kapıya çıkıyor. Aralarındaki farkları ortaya koymak için yazımızın bir sonraki bölümüne geçebiliriz.

### 3 Tmux ve Screen Arasındaki Farklar

Belirtilmiş olduğu üzere aslında işlev olarak Tmux ve Screen aynı amaca hizmet ediyorlar. Fakat aralarında bazı farklılıklar bulunmakta. Screen kullanan çoğunluk için bu farklar Tmux'a geçmeye yeter mi bilinmez, fakat etrafta Tmux'a geçenleri gördükçe ileride ortaya bir Gnome/KDE, Vim/Emacs tarzı bir rekabet çıkabileceğini söyleyebilirim. Ancak sayılan ikililer aynı amaçlara hizmet etse de hem yapı olarak hem de görünüş olarak birbirinden çok farklıdır. Tmux ve Screen ise bu kadar farklı görünmezler, en azından dışarıdan baktığımızda. Bu nedenle yazıyı okuyup da "Screen'den fazla nesi var ki?" diyenleri duyar gibi oluyorum :) Biz aradaki farkları aktaralım, karar yine kullanıcıların olsun.

- İki uygulamanın arasındaki en büyük farklardan biri lisanslarıdır. Screen, GPL ile dağıtılırken Tmux, BSD lisansını kullanır. Özellikle lisanslar konusunda çok aşırı hassas kullanıcılar için iki uygulama arasındaki bu farkın belirtilmesi gerekir.
- Tmux çalışma prensibi olarak Screen'den ayrılır. Tmux, Sunucu/İstemci modeliyle iş görür. İlk Tmux oturumu oluşturulduğunda eş zamanlı olarak bir de Tmux sunucusu oluşturulur. Bunun akabinde artık oluşturulacak tüm oturumlar bu sunucu için birer istemci olurlar. Tmux'un oturumları arasında iş görmek kolay ve düzenlidir. Burada öne çıkabilecek tek eksi puan ise sunucunun çökmesi durumunda tüm Tmux oturumlarının kaybedilecek olmasıdır.
- Screen \*nix dünyasındaki en yaygın çoklayıcıdır. Ayrıca girmedığı ortam kalmamıştır. 1987'den beri geliştirildiğinden dolayı kendisi kararlı sayılabilir de hala düzeltilmemiş sorunları mevcuttur. Kodu (bu işten anlayanların ifadesiyle) çorba gibi (spaghetti code) olmuştur. Bazı gereksiz sayılabilecek özellikler nedeniyle şişmiş durumdadır. Son sürümünü 2008'de çıkartmıştır ve yeni özellikler ekleme konusunda geliştiricileri pek hevesli değildir, örneğin pencereleri dikey bölmek gibi. Tmux henüz yeni yaygınlaşmaktadır. Kendisi IRIX ve HP-UX gibi ortamlarda desteklenmemektedir, bu ortamlar için destek verilmesi de düşünülmemektedir. Diğer taraftaysa Tmux, OpenBSD üzerinde 2009'dan beri öntanımlı çoklayıcı olarak yerini almıştır. Kodu yine bu işten anlayanların yorumuyla son derece düzenli ve güzel yazılmış durumdadır. Yeni olduğu için sürekli güncellenmekte ve yeni özellikler katılmaktadır.
- Tmux temiz ve hafif bir çoklayıcıdır. Zaten yazılmaya başladığında da bu yönde bir hedef belirlenmiştir. Screen'in neredeyse yarısı kadar bellek harcamaktadır. Çoklayıcıların bellek harcamaları fazla olmasa bile bu kullanım oranları uygulamaların yapısı hakkında fikir verecektir. Tmux kendi tecrübelerimden edindiğim kadarıyla da Screen'den daha hızlı cevap vermekte. Örneğin Screen'de verilecek bir dmesg komutunu ctrl+c ile kesmek Tmux ile yapılan denemedekinden çok daha yavaştır.
- Pencere yönetimi Tmux ile çok daha kolaydır. Pencerelerin boyutlandırılması için bir iki tuş kombinasyonu yeterlidir. Tmux bu konuda daha zekice davranmaktadır. Pencereleri dikey bölmek Tmux ile öntanımlı olarak sağlanırken, Screen'in bir yamaya yeniden derlenmesi gerekir. Belirtildiği üzere geliştiriciler bu özelliği eklemeyi düşünmemektedirler. Ubuntu üzerinde gelen Screen'se bu yama ile dağıtılmaktadır.
- Tmux oturumunda kullanabilen hemen her şey dışarıdaki kabukta da kullanılabilir. Bu özelliği sayesinde daha kolay Tmux betikleri yazılabilir. Tmux ile beraber durum satırını (çubuğu) öntanımlı olarak gelmektedir ve özelleştirmek çok daha kolaydır. Ayrıca kendisi bu konuda daha esnektir. Screen ise durum çubuğunu öntanımlı olarak getirmez ve .screenrc dosyasına işleyeceğimiz durum çubuğu satırı çok karmaşıktır.

Tmux'un öne çıkan bazı özellikleriyle Screen'den nasıl farklılaştığına daha da fazla değinebilirdim aslında, ancak listemiz kabarıp gidecektir. Bu nedenle aradaki farkları yüzeysel olarak maddeleyip diğer bazı farkları kullanarak veya araştırarak öğrenmeniz için size bırakıyorum. Yine de maddelemesek bile Tmux'un öne çıkan bazı özelliklerinden bahsedelim.

Tmux ile pencereleri istediğimiz bir anda eşzamanlı olarak kullanabiliriz. Şöyle düşünün. İki pencere var ve ayrı ayrı iki sunucuya bağlandık. Hatta sunucu sayısı iki değil de on olsun, haliyle pencere veya bölme sayımız da on olacaktır. Bu sunucuların hepsinde aynı anda apt-get update ve apt-get upgrade yürütmek istiyoruz. Tüm sunucu bağlantılarını yaptıktan sonra Tmux oturumunda bir komut yürüterek tüm bu pencereleri "senkronize" edebiliriz:

set—window—option synchronize—panes on|off

Birer adet apt-get update ve apt-get upgrade komutu on sunucuda birden çalışacaktır. Bu işlemten sonra birleştirdiğimiz pencere veya bölmeleri ayırarak tekrar her sunucu için ayrı ayrı çalışmaya devam edebiliriz.

Bu gibi küçük olarak görülebilecek bir çok özelliği sayesinde Tmux esneklik kazanmakta ve Screen'in önüne geçmekte. Yeni bir uygulama olduğundan, önünde de Screen gibi güzel bir örnek bulunduğu için Tmux muhtemelen Screen'in yaptıklarını yapmakla kalmayarak üzerine birçok şeyi koymakta. Bu gibi bir yazıda tüm bu farklılıkları anlatmak mümkün olmadığından ötürü bir sonraki bölümümüz olan Tmux'un kullanımına geçebiliriz.

## 4 Tmux'un Kullanımı

Bu kadar laf ebeliği yaptıktan sonra çok genel konuşmuş gibi görünüp, uygulamaların işlevleri hakkında pratik bilgi vermemiş olabiliriz. Fakat bu tür uygulamaları kullanmadan önce ne şekilde yararlanabileceğimizi öğrenmek veya kullanım alanlarını hayal edebilmek için ön bilgi şarttır. Bu aşamadan sonra tamamen kullanıma yönelik anlatımımıza geçebiliriz.

Anlatımın içinde boğulmamak için izleyeceğimiz yolu önce Bölüm 1'e göre belirleyelim ve oradaki dört aşama üzerinde sınırlama yapalım. Temel kaideleri öğrendikten sonra daha farklı konulara ve ipuçlarına değinmeye çalışalım. Başlamadan önce şunu belirtelim, Tmux'ta bir işlevi yerine getirmenin birden fazla yolu vardır. Anlatımda mümkün mertebe en pratik ve kolay olan yolları kullanmaya çalışacağız.

Ancak, devam etmeden önce çoklayıcıların en temel tuş vuruşundan bahsetmemiz şarttır. Çoklayıcılarda bir şeyler yapmak için önek tuşlarını kullanırız. Screen kullananların tahmin edebileceği gibi bu tuş Ctrl+a'dır. Tmux ise önek olarak Ctrl+b kombinasyonunu kullanır. Fakat anlatımlarda bu genelde Ctrl+b şeklinde geçmez. Ctrl-b daha da doğrusu C-b şeklinde geçmektedir. Bu nedenle bu yazıda da C-b gördüğümüz yerde Ctrl ve b tuşlarına aynı anda basarak önek tuşunu işlememiz gerekir. Bunun ardından diğer komut tuşları gelir. Örneklerini birazdan göreceğimiz için yolumuza devam edelim.

**Not:** Tmux sistemde kurulu olarak gelmez. Uygulama Merkezi'nden aratarak veya aşağıdaki komutla doğrudan kurabilirsiniz.

```
1 sudo apt-get install tmux
```

### 4.1 Oturum Yönetimi / Oturumdan Ayrılma, Geri Bağlanma / Oturum Kurtarma

Tmux'un (genel anlamda çokluyucuların) en güzel özelliklerinden birinin oturumları yönetebilmek olduğunu söylemiş-tik. Hemen elimizi kirlletmeye başlayabiliriz. Uçbirimi açtıktan sonra aşağıdaki komutlardan birini verirse karşımıza bir adet Tmux oturumu açılacaktır. İlk sıradaki standart bir oturum başlatmak için en kolay olanıdır görülebileceği üzere.

```
1 tmux
2 tmux new
3 tmux new-session
```

Karşımıza gelen şey, ilk esnada gözümüze farklı görünmeyebilir. Fakat en alt satırdaki yeşil durum çubuğu dikkatimizi çekecektir. Açılan bu ortam Tmux'un oturumdur. Durum çubuğunu sol tarafında "[0] 0:bash" ibaresi, sağ tarafında ise "sistemin\_ismi" "saat" "tarih" görünecektir. Sol taraftaki [0] oturumun ismi, 0:bash ise bu oturumdaki pencerenin ismidir. karakteri ise aktif pencereyi gösterir. Şu an için tek oturumumuz ve tek penceremiz olduğu için görüp göreceğimiz bu kadardır. Bunun haricinde oturumun Tmux'a ait olduğunu fark edecek başka bir belirti yoktur uçbirimde, tabi şu aşamada. Uçbirimde yaptığımız her şeyi burada da (haliyle) yapabiliriz. Deneme yapmaya başlayalım. less aracı ile /etc/fstab dosyamıza bakalım.

```
1 less /etc/fstab
```

Evet, önümüze dosyamız açıldı. Çokluyucuların en güzel özelliklerinden biri olan oturumu geri plana atıp, daha sonra bağlanmayı uygulayalım.

Sırasıyla aşağıdaki tuşları işleyelim:

```
1 C -b d
```

Bunun anlamı sırasıyla ctrl ve (ctrl'den elimizi kaldırmadan) b'ye aynı anda basıp, ardından iki tuşu birden bırakıp daha sonra d tuşuna basmaktır. Eğer tam olarak böyle yapmazsak istediğimiz şey olmaz. Doğru bir şekilde bunu uygularsak Tmux oturumu geri plana atılır ve uçbirimde "[detached]" yazısı görünür. Bunun anlamı oturumumuzun başarılı bir şekilde geri plana atıldığıdır. Biz farkında olmasak da geri planda less ile fstab dosyası açık olarak beklemektedir.



Oturumumuzu tekrar önümüze getirmek istiyoruz. Yapmamız gereken uçbirimde şu komutu yürütmektir:

```
1 tmux a
```

Bu komut aslında "tmux attach" komutunun kısaltılmış halidir. Bazı komutları kısa şekliyle kullanabiliriz, fakat hepsi için bu geçerli değildir. Komutu uyguladıktan sonra karşımıza geri plana atmış olduğumuz oturumumuz gelecektir. Bu oturum istersek uçbirimi kapatalım, istersek masaüstü oturumumuzu kapatalım tekrar bir uçbirim açıp çağırduğumuzda önümüze gelecektir. Hemen deneyelim bunu, yeni bir uçbirim açıp şu komutu yürütelim:

```
1 pkill gnome-terminal
```

Bu komutla bindiğimiz dalı kesiyoruz :) Komuttan sonra hem komutu uyguladığımız hem de Tmux oturumunun olduğu uçbirim sonlandırılacaktır. Bunu yapmamızın sebebi, uçbirimin bir şekilde zorla kapanması senaryosunu hayata geçirmektir. Zira uçbirimi çarpı tuşuyla kapatmaya çalışırsak bize aktif bir süreç olduğunu, eğer uçbirimi kapatırsak onun da sonlanacağını söyleyecektir. Bizim uyguladığımız komutla aynı işi görecektir fakat pkill ile sonlandırarak zorlama durumunda bile Tmux oturumunun kurtarılabilmesini görmek istiyoruz şu aşamada. Yeniden açacağımız uçbirimde tek yapmamız gereken "tmux a" komutunu yürütmektir. Oturumumuz yine karşımıza gelecektir. Bu şekilde ufak bir masaüstü çakılması veya uzak sunucudan bağlantımızın kopması senaryosunu hayata geçirmiş ve yaptığımız işi kırtarmış olduk.

less aracı ile dosyamıza bakarken oturumumuzda bir şey dikkatimizi çekecektir. Durum çubuğunun sol tarafında artık şu ibare yer almaktadır:

"[0] 0:less\*". Boş olarak açtığımız oturumdan farklı olarak less aracını kullandığımızda burada "bash" yerine artık "less" yazmaktadır. less aracını kapattığımızda yine "bash"e dönmektedir. Tahmin edebileceğimiz üzere Tmux, pencereleri kendiliğinden isimlendirmektedir. Örneğin less aracını "q" ile kapatıp uçbirimde "top" komutunu yürüttüğümüzde bu sefer isim "top" olarak değişecektir. Çoklu pencerelerle çalışmaya başladığımızda bunun yararını daha iyi göreceğiz.

Peki, yaptığımız işlemi Tmux algılayamadı ve pencere ismini değiştirmede ya da değiştirdi fakat biz farklı bir isim istiyoruz, bunu aşağıdaki tuşları işleyerek yapabiliriz:

```
1 C-b ,
```

Tuşların işlenmesinden sonra Tmux durum çubuğunda bizden pencere ismini düzeltmemizi isteyecektir. O an isim neyse mevcut şekilde tutulur, bizim bunu silip yerine istediğimiz ismi yazmamız gerekir. Bu isimlendirmeden sonra Tmux işlemlere göre kendiliğinden pencere isimlendirmesi yapmayacaktır. Bizim belirlediğimiz isim her neyse, sabit olarak kalacaktır.

Evet, standart bir oturum açmayı, oturumu geri plana almayı ve tekrar kendisine bağlanmayı gördük. Peki birden fazla oturuma ihtiyacımız olursa bu işlemler ne kadar değişecektir? Oturum açma komutunu her verdiğimizde Tmux yeni bir oturum oluşturacaktır. Bir uçbirim açıp "tmux" komutunu yürütelim, ardından bir başka uçbirim açarak bir "tmux" komutu daha yürütelim. İkincisine dikkat edersek eğer, durum çubuğunun solunda [0] yerine [1] ile başlayan bölümü göreceğiz. Anlaşılacağı üzere bu bizim ikinci oturumumuzun ismidir. Tmux'ta oturumlar 0'dan itibaren isimlendirilmeye başlarlar. Şimdi iki oturumu birden geri plana alalım. Artık bunu nasıl yapacağımızı biliyoruz: C-b d

İki oturumumuz da geri plana alındıktan sonra uçbirimlerden birini kapatabiliriz. Şimdi, uçbirimde şu komutu yürütelim:

```
1 tmux ls
```

Bu komut mevcut Tmux oturumlarını "listelememizi" sağlar ve çıktısı aşağıdaki gibidir:

```
1 0: 1 windows (created Sun Aug 14 13:26:07 2011) [142x42]
2 1: 1 windows (created Sun Aug 14 13:31:57 2011) [141x40]
```

Burada oturumlarımız listelenir ve kendileri hakkında bazı bilgiler mevcuttur. Oturum ismi, kaç pencereye sahip olduğu, oluşturulduğu tarih, boyutu gibi. İkinci oturumumuza bağlanmak istiyoruz. Uygulayacağımız uçbirim komutu şu olacaktır:

```
1 tmux a -t 1
```

“a” komutu “attach” ifadesini kısaltılmış haliydi hatırlayacak olursak. -t ise target-session anlamına gelir fakat target-session diye bir kullanım yoktur. “tmux a -t” ifadesinden sonra hangi oturumu ön plana almak istiyorsak o oturumun ismini veya şu haliyle daha doğru olan şekilde numarasını kullanmamız gerekiyor. Bazı durumlarda yaptığımız işin mevcut oturumumuzda görünmemesini, kendi halinde arka planda devam etmesini isteyebiliriz. İşte bu gibi durumlarda birden fazla oturum açıp kullanmak durumunda kalırız. Fakat böyle olduğunda yukarıdaki listeden görebileceğimiz üzere ufak bir karışıklık yaşayabiliriz: Hangi oturumda ne vardı? Bunun önüne geçmek için oturumlarımızı isimlendirerek açabiliriz. Diyelim ki bir oturumumuzda “derleme” işi devam edecek, diğer oturumumuzda da “genel” işlerimizi yapacağız.

Oturumlarımızı açmak için şu iki komutu kullanabiliriz:

```
1 tmux new -s genel
2 tmux new -s derleme
```

İki oturumu da arka plana aldıktan sonra “tmux ls” komutumuz şuna benzer bir çıktı verecektir bize:

```
1 derleme: 1 windows (created Sun Aug 14 13:54:30 2011) [80x23]
2 genel: 1 windows (created Sun Aug 14 13:54:26 2011) [80x23]
```

Görebileceğimiz üzere artık oturumlarımızın birer ismi var. Bu oturumlardan birine tekrar bağlanmak için tahmin edebileceğimiz üzere şu komutu kullanıyoruz:

```
1 tmux a -t derleme
```

Burada hemen bir ipucu verelim. Burada yaptığımız gibi iki oturum için iki ayrı uçbirim kullandık ya da önce birini açıp, geri plana alıp, ardından ikincisi için yeni komutumuzu verdik. Bunun yerine doğrudan arka plana alınmış bir oturum da oluşturabiliriz.

```
1 tmux new -d -s genel
```

Bu şekilde oturumumuz açılır ve arka plana atılır. Hatta oturumumuzu penceresine isim vererek bile açabiliriz:

```
1 tmux new -d -n pencere_ismi -s genel
```

Oturum yönetimi oldukça esnek bir Tmux’ta ancak daha fazla detaya inerek hali hazırda uzamış olan bu yazıyı daha fazla uzatmayalım, zira değinilecek daha temel meseleler bizleri bekliyor. Bu ve benzeri birçok özellik ve işlevi gerek Tmux’un “man” bölümünde, gerekse internette rahatlıkla bulabiliriz. Fakat Türkçe kaynak bulmakta zorlanabiliriz, hatta korkum o ki Tmux için tek Türkçe kaynak şu an okuduğumuz yazı bile olabilir :)

## 4.2 Pencere Çoklama / Yönetme

Artık bir Tmux oturumunu nasıl oluşturacağımızı, gerektiğinde nasıl geri plana alacağımızı ardından da nasıl tekrar bağlanabileceğimizi öğrendik. Şu andan itibaren biraz daha işimize yarayacak olan pencerelerle iş görmeye başlayabiliriz. Pencere yönetimi çoklayıcılardaki en pratik işlevlerden biridir.

Eğer elimizin altında bir oturum yoksa hemen fişekleyelim ve karşısına geçelim. Oturumumuza isim verdiğimizde durum çubuğunda sol başta bu isim, devamındaysa pencere numarası ve ismi görünecektir. Şu ana kadar pencere çoklamayı görmediğimiz için burada bir adet penceremiz bulunur. Eğer bir önceki bölümde verilen ipucunu kullandıysak veya “C-b,” ile penceremize isim verdiğimizde kendi ismi, vermediyse burada “0:bash\*” yazıyor olacaktır. Kısacası şu an bir adet penceremiz mevcuttur. Tmux oturumunda yeni pencereler oluşturmak için aşağıdaki tuş ikilisini işleriz:

```
1 C-b c
```

Bu işlemden hemen sonra yeni bir pencere açılacak ve aktif olarak kendimizi o pencerede bulacağız. Bu arada durum çubuğuna dikkat edersek pencere sayısının ikiye çıktığını göreceğiz: "0: bash- 1: bash\*" Daha önce "\*" karakterinin anlamını belirtmiştik, bu aktif pencere anlamına gelir. "-" karakteri ise "bir önceki" pencere anlamına gelir. İki pencere ile "bir önceki" ifadesi pek anlam ifade etmiyor olabilir, ancak pencere sayımız arttığında bunun ne olduğunu daha iyi anlayacağız.

Dilediğimiz kadar pencere açıp, bunlara "C-b ," ile isimler verebiliriz. Örneğin birinde "top" komutunu yürütürken diğerinde kopyalama yapabilir, bir başkasında dosyaları inceleyip yapılandırmalarımızı düzenleyebiliriz. Peki pencereler arasında nasıl geçiş yapacağız?

Daha önce söylendiği üzere Tmux'ta bir işi yapmanın birden fazla yolu vardır. Aralarından hangisini kullanacağımızı tamamen bize bağlıdır. Hangisi bize daha pratik geliyorsa, hangisini daha hızlı kullanabiliyorsak onu seçebiliriz. Bu yollardan birkaçını ifade etmeye çalışalım.

Pencereler arasında ileri geri gitmek için:

- 1 C-b n
- 2 C-b p

işlemlerini yürütebiliriz. n bir sonraki, p ise bir önceki pencereye gidecektir.

Pencereleri numarasına göre seçim yapmak için:

- 1 C-b "pencere numarası"

kullanılır. Yani ikinci pencereye geçmek için:

- 1 C-b 2

yapmamız yeterlidir. Eğer pencere sayımız 10'dan fazlaysa

- 1 C-b '

işlemini kullanabiliriz. Bizden bir "index" isteyecektir, istediğimizi yazıp penceremize geçebiliriz.

Pencereler arasında geçiş yapmanın bir başka pratik yoluysa

- 1 C-b w

işlemini kullanmaktır. Bunu yaptığımızda karşımıza pencerelerimizin bir listesi çıkacaktır. Numarasını seçerek geçiş yapabileceğimiz gibi klavyemizin yön tuşlarını kullanarak da dilediğimiz pencereye atlayabiliriz. Eğer pencere sayımız yine 10'dan fazlaysa numaralar 10'dan sonra yerini harflere bırakır. a,b,c şeklinde devam ederler ve biz numara yerine 10. pencere için a tuşuna basarız.

Bu aşamada bölümün girişinde bahsetmiş olduğumuz pencere isimlerinin yanında görünen "-" karakterini açıklayalım. "-" söylediğimiz gibi bir önceki pencere anlamına gelir. Örneğin 1. penceredeydik ve "C-b 2" ile üçüncü penceremize geçtik. Durum çubuğu şu şekilde görünecektir:

- 1 0: bash- 1: bash 2: bash\*

3. penceremiz aktif durumdadır, 1. penceremiz ise "bir önceki" pencere konumundadır. Klasik olarak pencereler arasında geçiş yapmak yerine istediğimiz iki pencere arasında:

- 1 C-b l

ile ileri geri yapabiliriz. Böylece aradaki pencereler atlanacak ve bizim iki penceremiz arasında geçiş yapılacaktır.

Pencere sayımız çok olduğunda ortalık görülebileceği üzere biraz karışabiliyor. Pencereleri listeleterek geçiş yapmak mümkün, fakat bunun yerine isim verdiğimiz pencereler arasında arama yaparak da geçiş yapabiliriz. Pencerelerimize yaptığımız işlemlerle alakalı isimler verirsek aramalarımızda rahat edebiliriz. Örneğin "derleme", "kopyalama".

1 C-b f

ile aramayı tetiklediğimizde durum çubuğu üzerinde Tmux bizden isim girmemizi isteyecektir. İsmi girdiğimizde eğer benzer isimde başka bir pencere yoksa doğrudan oraya atlayacaktır. Bizim örneğimizden yola çıkarsak arama bölümüne "der" girsek bile "derleme" isimli pencereye geçiş yapacaktır Tmux. Eğer benzer isimde pencereler varsa karşımıza bir liste gelecektir ve bizden seçim yapmamızı isteyecektir. Ancak burada bir şeyi belirtmekte fayda var. Tmux aramalarını sadece pencereler üzerinde yapmaz, pencerelerin o an gördüğü işlev neyse bunun üzerinden arama yapar. Mesela bir pencere sadece uçbirimde bekliyor ve bulunduğu dizinde önceden "ls" çalıştırılmışsa, listelenen dizin ve/veya dosya isimlerinde de örneğin "kitap" geçiyorsa, arama buraya kadar genişleyecektir. Kısacası biz "kitap" yazdığımızda bu pencereye geçiş yapılacaktır. Ufak bir not da aramanın büyük/küçük harf duyarlı olduğu üzerine olsun.

Pencerelerle iyice haşır neşir olduğumuza göre artık bölmelerimize geçebiliriz. Çoklayıcıların güzel özelliklerinden biri de pencerelerini bölebiliyor olmamızdır. Tmux üzerinde pencereleri öntanımlı olarak hem dikey hem de yatay konumda bölebiliriz.

#### **Dikey bölmek için:**

1 C-b %

#### **Yatay bölmek için:**

1 C-b "

işlemlerini yürütmemiz yeterli olacaktır. Bu şekilde istediğimiz bir pencereyi bölebilir, hatta bölmelerimizi de tekrar bölebiliriz. Bölmelerle çalışmak tamamen bizim ihtiyacımıza ve hayal gücümüze kalıyor. Daha önce de belirtmiş olduğumuz gibi, bir bölmede dosyamızı düzenlerken, bir başka bölmede sunucuyu "top" ile izleyebilir, bir başka bölmede IRC sunucusu üzerinde sohbet edebiliriz. Hatta Tmux'u sunucuda değil de kendi yerelimizdeki bir sistemde kullanıyorsak bir başk bölmede uçbirimden çalışan bir oynatıcı ile müzik bile dinleyebiliriz.

Evet, pencerelerimizi bölmelere ayırmayı öğrendik. Peki bu bölmeler arasında nasıl geçiş yapabiliriz? Her zamanki gibi bunun da birden fazla yolu mevcut. Sırasıyla değinmeye çalışalım.

Bölmeler arasında en basit geçiş yolu belki de

1 C-b "yön tuşları"

işlevidir. Ancak burada bir ayrıntıyı belirtmemiz lazım. Bu işlemi kullanmak için "Ctrl" ve "b" tuşuna basıp, bıraktıktan sonra yön tuşlarını kullanmalıyız. Aksi halde bırakmadan geçiş yapmaya çalışırsak yerel makinemizde farklı bir özelliği tetikleyecektir, ileride anlatacağımız bir özellik bu. Uzak bir makinede çalışıyorsak böyle bir şey sözkonusu değildir. Yön tuşlarının dördünü de kullanabiliriz, kısacası aşağı,yukarı ve sağa,sola geçiş yapmamız mümkündür.

Bölmeler arasında gezinmenin bir başka yolu ise

1 C-b o

işlemi ile sırayla (ileri doğru) bölmeler arasında gezinmektir. Geri doğru gezinme mümkün olmasa bile bir önceki bölmeye geçiş aşağıdaki işlemle yapılabilir.

1 C-b ;

Bölmeler arasında gezinmenin ilginç bir başka yolu ise şu işlemi yürütmektir:

1 C-b q

Bunu yaptığınızda bölmelerin numarası, üzerlerinde büyükçe bir şekilde belirecektir. Aktif bölme kırmızı, pasif bölme veya bölmelerse mavi renkle temsil edilirler. Ancak bu numaraların belirme süresi kısadır ve hızlı bir şekilde seçim yapmamız gerekir.

Bölmelerimizi oluşturmayı öğrendik. Fakat bir bölme ihtiyacımızdan daha fazla yer işgal ediyor ve biz bunu daha dar bir alana yaymak istiyoruz ya da tam tersi. Bu durumda yapabileceğimiz şey bölmelerimizi boyutlandırmaktır. Evet, bunun da birden fazla yolu vardır ve yereldeki makinemizle uzaktaki makinemiz arasında farklılıklar gösterecektir.

Aslında mesele yerel veya uzak makine meselesinden ziyade bir “uçbirim öykünücüsü” kullanıp kullanmama meselesidir. Masaüstünüzde gnome-terminal kullandığımız için boyutlandırma işi biraz daha rahat yapılabilir. Ancak uzak bir sunucuya bağlandığımızda “uçbirim öykünücüsü” yerine uçbirimin kendisini (daha doğrusu sanal uçbirimi) kullandığımız için biraz farklılık yaşayacağız.

Önce gnome-terminal üzerindeki kolay yoldan bahsedelim. Bir bölmeyi boyutlandırmak için:

```
1 C-b C-"yukarı , aşağı , sağ , sol"
```

işlemini kullanabiliriz. Yani Ctrl+b'ye basıp, bırakıp Ctrl+"yön tuşlarını" kullanacağız. Bu şekilde yaparak bölmemizi dilediğimiz gibi boyutlandırabiliriz. Fakat hemen belirtelim bu son boyutlandırma işleminin de bir süresi var. Kısacası bölmemizi Ctrl+"yön tuşları" ile hareket ettirirken bu hareketi ara vermeden yapmamız lazım.

gnome-terminal değil de sanal bir uçbirim kullandığımızda, yani uzak sunucuya bağlandığımızda yön tuşları ne yazık ki işe yaramayacaktır. Burada elimizi biraz kirlitememiz gerekecek. Bu aşamadan sonra hem Tmux'un komut satırını göreceğiz hem de tuş ataması yapmayı öğreneceğiz.

Tmux ile çalışırken

```
1 C-b :
```

işlemini kullanırsak bizi kendi komut satırına düşürür ve bizden bir komut bekler. Bu komut satırı durum çubuğunun yerini alacaktır. Denememizi yapalım. Bu komut satırına şunu yazalım:

```
1 split
```

ardından TAB tuşuna basalım. Kendisi bu komutu split-window olarak tamamlayacaktır. “Enter” tuşuna bastığımızda ise aktif bölmeyi veya pencereyi yatay olarak bölecektir. Buradan da anlayabileceğimiz gibi Tmux'a bazı işlevlerini kendi komut satırını kullanarak yaptırabiliyoruz ve bu komut satırında aynen kendi uçbirimimizde olduğu gibi komut tamamlama özelliği bulunuyor.

Şimdi, bu komut satırını kullanarak bölmelerimizi boyutlandıralım. Bölmeleri boyutlandırmak için “resize-pane” komutunu ve bu komutun parametrelerini kullanacağız. Örneğin bölmeyi aşağı doğru boyutlandırmak için:

```
1 resize-pane -D
```

komutunu yürütebiliriz. -D aşağı, -U yukarı, -R sağ ve -L sol boyutlandırma için kullanılır. Öntanımlı olarak ölme 1 birim kaydırılır. Farklı birimlerle iş görmek istiyorsak:

```
1 resize-pane -D 5
```

gibi bir komut uygulayabiliriz. Peki her seferinde bu komutu yazmak zor olmayacak mı? Bu zorluğu aşmak için bir tuşu bu komuta atamamız mümkün. Bunun için komut satırına örneğin k tuşunu aşağı boyutlandırma için atamak için şunu yazabiliriz:

```
1 bind-key k resize-pane -D
```

Artık

```
1 C-b k
```

yaptığımızda bölmemiz aşağıya doğru boyutlandırılacaktır. Kısayol tuş atamalarında Türkçe harf kullanmamaya gayret göstermeliyiz. Tuş atamalarında ayrıca sadece harfler yerine Ctrl ve Alt önekli harfleri de kullanabiliriz. Örneğin:

```
1 bind-key M-a k resize-pane -D
```

ile önce Alt-a'ya bu atamayı yapabiliriz. "M" harfi buada Alt tuşu için kullanılır. Ctrl için "C" harfini kullanabiliriz.

Fakat bu tuş bağlama işlemi sadece açık oturumumuz için geçerli olur, oturumumuzu sonlandırdığımızda ayarlamamız uçup gidecektir. Kalıcı ayarlarımızın olması için Tmux'un kullandığı yapılandırma dosyamıza işlemler yapmak zorundayız.

Tmux yapılandırma dosyası olarak ev dizinimizdeki gizli .tmux.conf dosyasını kullanır. Eğer kurulumdan sonra bu dosya oluşmadıysa kendimiz oluşturabiliriz. Bu noktada Tmux'a Screen'den geçecekler için bir öneride bulunabiliriz. Screen önek tuşu olarak Ctrl+a kullandığı için Tmux'a geçenlerin eli sürekli kendisini arayabilir. Dileyen olursa Tmux'un kullandığı Ctrl+b önek tuşunu Ctrl+a olarak ayarlayabilir.

Tmux'un yapılandırma dosyası daha düzenlidir. Biraz araştırmayla rahatlıkla işimizi görebilecek atamaları yapmayı öğrenebiliriz. Önek tuşunu değiştirme örneği ile başlayalım. Dosyamıza aşağıdaki değişikliği eklediğimizde önek tuşumuz artık C-a olacaktır:

```
1 set -g prefix C-a
```

Tmux başlarken bu dosyadan ilgili değişiklikleri okur. Peki ya açık olan bir Tmux oturumu için değişiklik yaparsak kendisini kapatıp açmamız mı gerekir? Hayır. İlgili değişikliğin hemen etkin olması için şu komutu uygulamalıyız:

```
1 source-file ~/.tmux.conf
```

Bu dosyayı kullanarak dilediğimiz komutu boşta olan bir tuş takımına atamamız mümkün. Sadece tuş atamaları için değil, Tmux'un birçok ayarının başlangıçta bizim istediğimiz şekilde etkin veya kapalı olması için de bu dosyaya müracaat etmemiz gerekir.

Unuttuğumuz bir şeyi aktaralım. Pencere veya bölmeleri kapatmak için uçbirimdeki gibi "Ctrl+d" kullanmamız veya "exit" yazmamız yeterlidir. Tmux'un kendi tuş işlevi olan "C-x" ile kapatabilmemiz de mümkündür. Fakat bu işlev bizden pencere veya bölmeyi kapatmak isteyip istemediğimize dair bir onay isteyecektir. "y/n" tarzında bir soru olduğundan ve mutlaka bizden cevap beklediğinden dolayı pratikliğini biraz kaybetmiş fakat güvenliği bir adım öteye geçirmiş durumdadır.

Bölmelele ve pencerelerle ilgili birçok başka konu mevcuttur. Ancak yine yazımızın çok uzadığını fark ettiğimizden dolayı üzülerek diğer ayrıntıları kullanıcılara bırakıyoruz.

### 4.3 Oturum Paylaşma

Çokluyıcıların faydalarını sıralarken oturum paylaşmadan bahsetmiş ve azak bir yerde bir başkası ile beraber çalışırken yararlı olabileceğine değinmiştik. Şimdi bunu kısa bir şekilde anlatmaya çalışalım. Elimizin altında "genel" isminde bir oturum olduğunu varsayalım. Uzaktaki kişi bizim makinemize SSH ile bağlandıktan sonra yapması gereken tek şey:

```
1 tmux a -t genel
```

komutunu vermektir. Bu aşamdan sonra iki tarafın da yapacağı her şey birebir iki pencerede de aynı olacaktır. Daha önce söylediğimiz gibi sohbet etmek bile mümkündür. Uzaktaki kullanıcı ayrılacağı zaman dikkat edeceği şeyse çıkış yapmak yerine "C-b d" ile oturumu kendi adına geri plana atmaktır. Daha sonra sizin sisteminizden de ayrılabilir. Eğer çıkış yaparsa oturum ortak olduğu için siz de çıkış yapmış olursunuz.

Burada küçük ancak işe yarar bir ayrıntıyı aktarmakta fayda var. Uzaktaki kullanıcı bizim oturumumuza bağlandığında bir şeyi fark ederiz. Her kim pencere değiştirirse diğer kullanıcının oturumunda da pencere değişikliği olacaktır. Yani bir kullanıcı 2. pencereye geçerse diğer kullanıcı da bu pencereyi görecek. Oturum ortak olduğu için bu gayet normaldir. Fakat ya kullanıcılar tek pencerenin ortak olmasını, diğer pencerelerde bağımsız çalışabilmeyi isterlerse? Bu durumda uzaktaki kişi bizim oturumumuza şu şekilde bağlanmalıdır:

```
1 tmux new -t genel
```

Bu şekilde iki kullanıcı da ayrı ayrı çalışabilir. Ancak bir yanlış anlaşılma olmasın. Ayrı ayrı çalışırlar fakat aslında pencereler yine ortaktır. Kısacası birinci kullanıcı ikinci kullanıcının iş gördüğü pencereye geçerse yine yaptıklarını görebilecektir. Bağımsız olan farklı pencereleri görerek çalışabilmeleridir.

#### 4.4 Durum Çubuğunu Özelleştirme / Kopyalama,Yapıştırma Kipini Kullanma

Durum çubuğunun Tmux ile beraber öntanımlı olarak geldiğinden bahsetmiştik. Yine durum çubuğunu özelleştirmenin kolay olduğundan da dem vermiştik. Şimdiyse bir iki ufak değişiklikle bunu örnekleyelim. Durum çubuğunun rengi öntanımlı olarak yeşil ve yazı rengiyse siyahtır. İlk olarak bunu değiştirmeye başlayabiliriz işe.

```
1 set -g status-bg black
2 set -g status-fg white
```

komutlarını sırayla uyguladığımızda durum çubuğunun rengi siyah, yazılar ise beyaz olacaktır. Biraz da buradaki bilgilerle oynayalım. Şu aşamda sol tarafta oturumun ismi, pencerelerin numarası ve ismi gösterilmektedir. Bunu makine\_ismi/oturum ismi şeklinde yazdırıp, yazı rengini de yeşil yapmak için şunu kullanabiliriz:

```
1 set -g status-left '[fg=green]#H/#S'
```

Aktif olan pencereyi kırmızı renkli olarak görmek istiyoruz:

```
1 set-window-option -g window-status-current-bg red
```

Hareket olmasını beklediğimiz bir pencere var. Örneğin bir dosyanın son satırlarını "tail" ile takip ediyoruz ve yeni bir satır eklendiğinde haberdar olmak istiyoruz. Ya da IRC üzerinde bir yerlerde sohbet ediyoruz ve kanal geneline ya da özelimize bir ileti geldiğinde bilmek istiyoruz, hem de başka bir penceredeyken? Bunun için şunu kullanabiliriz:

```
1 setw -g monitor-activity on
```

Bu ayarlardan sonra hareketin olduğu pencere başlığı renk değiştirerek beyaz olacaktır. Ancak, bir de durum çubuğunda anlık bir ileti görmek istiyorsak şunu uygulamalıyız:

```
1 setw -g visual-activity on
```

Bu ayarlardan sonra kısa bir süreliğine hangi pencerede "aktivite" olduğu bize bildirilecektir.

Görülebileceği üzere durum çubuğunu özelleştirme çok kolay ve aynı zamanda esnektir. Şimdi de sağ tarafı biraz özelleştirelim. Öntanımlı olarak sağ tarafta makine ismi, saat ve tarih görülür. Biz yine yeşil renkli olarak burada sistemin yük değerini görelim.

```
1 set -g status-right '[fg=yellow]#{uptime | cut -d " , " -f 3}'
```

komutu ile bu işimizi görebiliriz. Sağ tarafın güncelleme süresi 15 saniyedir. Bunu değiştirmek için:

```
1 set -g status-interval 10
```

komutunu kullanabiliriz. Fark edebileceğimiz üzere durum çubuğunun sağ tarafını özelleştirmek için bash komutlarını rahatlıkla kullanabiliyoruz.

Durum çubuğu ile ilgili özelleştirmeleri kabaca görmüş olduk, detaylar yine kullanıcılara kalıyor. Şimdiyse çoklayıcıların bir başka özelliği olan kopyalama ve gezinme kipinden bahsedelim. Zira uzak bir sunucuda çalışırken bu özellik bizim için oldukça faydalı olacaktır.

Çoklayıcılarla çalışırken fark edeceğimiz üzere çıktılar uçbirimi doldurduktan sonra Page-up ile yukarı doğru gezinmeyiz. Bu bir dezavantajdır ancak üstesinden gelinemeyecek bir durum değildir. Kaldı ki bu durumun üstesinden gelirken uzak sunucularda çalıştığımızda yapamadığımız kopyalama işini de halletmiş oluruz.

Tmux ile yukarı satırlara doğru gezinmeye başlamak için kopyalama kipine geçmemiz lazım. Bunun içinse

```
1 C-b [
```

işlemimizi yürütmeliyiz. Bu aşamadan sonra klavyemizdeki yön tuşlarıyla yukarı, aşağı ve sağa, sola doğru gezinebiliriz. Kopyalamak istediğimiz yere başına geldiğimizde C-"boşluk" tuşuna basıp istediğimiz yeri işaretleyebiliriz. İşaretlememiz bittiğinde Alt+w tuşlarını kullandığımızda Tmux kopyalama kipinden çıkıp yine uçbirime düşecektir. Kopyaladığımız bölümü yapıştırmak içinse

```
1 C-b ]
```

işlemimizi yürütebiliriz.

Yazımızı burada noktıyoruz. Tmux hakkında daha yazılacak çok şey olmasına rağmen böyle bir yazıda tümüne değinmek mümkün değildir. Gerek kendisinin man bölümünden, gerek sitesinden, gerekse internetteki kaynaklardan hemen hepsi hakkında güzel örnekler bulmamız mümkündür. Bu yazıda benim eksik bıraktığım, yanlış ifade ettiğim, yanlış bildiğim ve aktardığım bölümler de olabilir. Böyle bir durumda ufak bir araştırmayla doğrusunu bulabileceğinizden eminim. Ancak yazıda belirttiğim üzere bu konuda Türkçe kaynak olmasını pek ummuyorum. Bu güzel uygulama hakkında ilk Türkçe kaynak olması dileğiyle.

**Not:** Bu anlatımda kabuk olarak **bash**, uçbirim olarak da **gnome-terminal** kullanılmıştır.