



LINUX 101 HACKS

Practical Examples to Build a
Strong Foundation in Linux

Ramesh Natarajan
www.thegeekstuff.com

İÇİNDEKİLER

Giriş.....	7
Önsöz	8
Baskı	9
Bölüm 1: Güçlü CD Komutu İpuçları.....	9
İpucu 1. CDPATH kullanarak cd komutu için anadizin tanımlamak.....	9
İpucu 2. Dizinlerde etkili şekilde gezinmek için cd alias'ı kullanma....	10
İpucu 3. mkdir ve cd'yi tek bir komutta uygulamak.....	12
İpucu 4. Son iki dizin arasında geçiş yapmak için “cd -” kullanmak.....	12
İpucu 5. dirs , pushd ve popd kullanarak dizin yığınlarını idare etme..	13
İpucu 6. “shopt -s cdspell” kullanarak cd'de yanlış yazılmış dizin adını otomatik olarak düzeltme	15
Bölüm 2: Tarih Yönetimi	15
İpucu 7. Sistemin tarih ve saatini ayarlama.....	15
İpucu 8. Donanımın tarih ve saatini ayarlama.....	16
İpucu 9. Şuanki tarihi ve saati özel formatta gösterme.....	17
İpucu 10. Geçmiş bir tarihi ve saati gösterme.....	18
İpucu 11. Gelecek bir tarihi ve saati gösterme.....	18
Bölüm 3: SSH İstemci Komutları.....	19
İpucu 12. SSH istemci versiyonunu öğrenmek.....	19
İpucu 13. SSH ile uzak sunucuda oturum açma.....	19
İpucu 14. SSH istemci oturumunda hata ayıklama.....	21
İpucu 15. SSH kaçış karakterleri ile SSH oturumunda geçiş yapmak...22	
İpucu 16. SSH kaçış karakterleri ile SSH oturum istatistikleri.....	23
Bölüm 4: Başlıca Linux Komutları.....	24
İpucu 17. Grep komutu.....	24
İpucu 18. Find komutu.....	25

İpucu 19. Standart çıktı ve hata mesajlarını yönlendirmek.....	27
İpucu 20. Join komutu.....	27
İpucu 21. Harf değiştirme.....	28
İpucu 22. Xargs komutu.....	28
İpucu 23. Sort komutu.....	29
İpucu 24. Uniq komutu.....	31
İpucu 25. Cut komutu.....	32
İpucu 26. Stat komutu.....	33
İpucu 27. Diff komutu.....	34
İpucu 28. Kullanıcıların toplam bağlı kalma sürelerini gösterme.....	34
Bölüm 5: PS1, PS2, PS3, PS4 ve PROMPT_COMMAND.....	36
İpucu 29. PS1 - Varsayılan Komut İstemcisi.....	36
İpucu 30. PS2 - İkincil komut istemcisi.....	37
İpucu 31. PS3 - Kabuk betiğinde “select” tarafından kullanılan satır....	37
İpucu 32. PS4 - set -x kullanıldığında çıktının ön ekini düzenleme.....	39
İpucu 33. PROMPT_COMMAND.....	40
Bölüm 6: PS1 kullanarak renkli ve fonksiyonel komut satırı.....	40
İpucu 34. Komut istemcisinde kullanıcı, bilgisayar ve dizin adı.....	40
İpucu 35. Komut istemcisinde şuanki saati gösterme.....	40
İpucu 36. Komut istemcisinde herhangi bir komutun çıktısını gösterme.....	41
İpucu 37. Komut istemcisinde yazıların rengini değiştirme.....	41
İpucu 38. Komut istemcisinde arkaplan rengini değiştirme.....	42
İpucu 39. Komut istemcisinde çoklu renk gösterme.....	43
İpucu 40. tput kullanarak komut istemcisinde renk değiştirme.....	44
İpucu 41. PS1 değişkeni için mevcut kodları kullanarak kendi komut istemcisini yaratma.....	45
İpucu 42. PS1 değişkeni içerisinde bash kabuk fonksiyonu kullanma..	46
İpucu 43. PS1 değişkeni içerisinde kabuk betiği kullanmak.....	46

Bölüm 7: Arşiv ve Sıkıştırma.....	47
İpucu 44. Temel zip komutları.....	47
İpucu 45. Zip komutu kullanarak gelişmiş sıkıştırma.....	49
İpucu 46. Şifre korumalı zip dosyaları.....	49
İpucu 47. Zip arşivi kontrol etmek.....	50
İpucu 48. Temel tar komutları.....	50
İpucu 49. gzip ve bzip2'yi tar ile birleştirme.....	51
Bölüm 8: Komut Satırı Geçmişi.....	52
İpucu 50. HISTTIMEFORMAT kullanarak geçmişte komutun zamanını göstermek.....	52
İpucu 51. Ctrl+R kullanarak geçmişte arama yapma.....	52
İpucu 52. Dört farklı yöntem kullanarak hızlı bir şekilde önceki komutu tekrar etme.....	53
İpucu 53. Geçmişten özel komut çalıştırma.....	54
İpucu 54. Özel kelime ile başlayan önceki komutu çalıştırma.....	54
İpucu 55. HISTSIZE kullanarak geçmişteki toplam satır sayısını kontrol etme.....	54
İpucu 56. HISTFILE kullanarak geçmiş dosyası adı değiştirme.....	55
İpucu 57. HISTCONTROL kullanarak geçmişte sürekli tekrarlanan girdileri elemek.....	55
İpucu 58. HISTCONTROL kullanarak geçmişteki tekrarlanmış komutları silmek.....	56
İpucu 59. HISTCONTROL ile özel bir komut kullanarak zorunlu olarak geçmişi hatırlamamak.....	56
İpucu 60. -c parametresi ile tüm geçmişi temizlemek.....	57
İpucu 61. Geçmiş komutun parametresini yeni bir komuta verme.....	57
İpucu 62. Geçmiş komutun istenilen parametresini yeni komuta vermek.....	57
İpucu 63. HISTSIZE kullanarak geçmişi devredışı bırakmak.....	58

İpucu 64. HISTIGNORE kullanarak özel komutların geçmişte kayıt edilmemesini sağlamak.....	58
Bölüm 9: Sistem Yöneticisi Görevleri.....	59
İpucu 65. fdisk ile diski bölümlenme.....	59
İpucu 66. mke2fsk ile diski biçimlendirme.....	60
İpucu 67. Bölümlenmiş diski mount etme.....	62
İpucu 68. tune2fs ile diske ince ayar yapma.....	62
İpucu 69. Takas alanı oluşturma.....	64
İpucu 70. Yeni kullanıcı oluşturma.....	64
İpucu 71. Yeni grup oluşturma ve ona bir kullanıcı atama.....	66
İpucu 72. OpenSSH'de SSH parolasız oturum açma ayarları.....	66
İpucu 73. ssh-agent ile birlikte ssh-copy-id kullanımını.....	67
İpucu 74. Crontab.....	69
İpucu 75. Magic SysRq Key kullanarak Linuxu güvenli yeniden başlatma.....	70
Bölüm 10: Apachectl ve Httpd Örnekleri.....	72
İpucu 76. Farklı httpd.conf dosyasından apachectl'ye geçme.....	73
İpucu 77. DocumentRoot'u httpd.conf dosyasını değiştirmeden geçici olarak kullanma.....	73
İpucu 78. Geçici olarak Log Level'i artırma.....	74
İpucu 79. Apache içindeki modülleri gösterme.....	75
İpucu 80. httpd.conf içerisinde kabul edilmiş şartları gösterme.....	76
İpucu 81. Değişiklik yaptıktan sonra httpd.conf'u doğrulamak.....	76
İpucu 82. httpd yapılandırma parametrelerini gösterme.....	77
İpucu 83. Sadece istenildiğinde özel modülü yüklemek.....	78
Bölüm 11: Bash Kabuk Betiği	79
İpucu 84. .bash_* dosyalarını çalıştırma sırası.....	79
İpucu 85. Bash kabuğunda rastgele sayı üretmek.....	80
İpucu 86. Kabuk betiğinde hata ayıklama.....	80

İpucu 87. Kaçış işaretleri ve tırnaklar.....	81
İpucu 88. Kabuk betiği içerisinde veri dosyası alanı okuma.....	83
Bölüm 12: Sistem Denetleme ve Performans.....	84
İpucu 89. Free komutu.....	84
İpucu 90. Top komutu.....	85
İpucu 91. Ps komutu.....	87
İpucu 92. Df komutu.....	89
İpucu 93. Kill komutu.....	90
İpucu 94. Du komutu.....	91
İpucu 95. lsof komutları.....	92
İpucu 96. Sar komutu.....	94
İpucu 97. Vmstat komutu.....	96
İpucu 98. Netstat komutu.....	97
İpucu 99. Sysctl komutu.....	99
İpucu 100. Nice komutu.....	100
İpucu 101. Renice komutu.....	101

Tanıtım

“Dünyada yalnızca 10 çeşit insan vardır. İkili(binary) kodu anlayanlar, anlayamayanlar ve [gray code](#)'u anlayanlar”

- [Geek](#)

Bu kitapta Linux'ta sağlam bir temel oluşturmanıza yardım edecek toplamda 101 ipucu var. Kolay takip etmeniz açısından tüm ipuçları Linux kullanılarak (örneklendirilerek) açıklandı.

Bu kitap 12 bölüm içeriyor. İpucu içeren 6 bölüm , zaten blog'umda yayınladığım makalelerden alınmıştır. Geri kalan 6 bölümde bahsedilen ipuçları ise yepyeni ipuçlarıdır.



Ben Ramesh Natarajan, The Geek Stuff blog'u ve bu e-kitabın yazarı.

Birkaç dilde programlama ile yoğun bir şekilde uğraştım ve favorim C dili. Altyapı tarafı dahil olmak üzere, Linux sistem yöneticiliği , DBA (DataBase Admin), Networking , Donanım ve Depolama alanlarında bir sürü iş yaptım.

Ayrıca Windows, Linux ve Mac'de çalışan ücretsiz, kolay ve güvenli şifre yöneticisi olan [Password Dragon](#)'u geliştirdim.

Eğer bu e-kitap hakkında ulaşmak isterseniz [buraya](#) tıklayabilirsiniz.

Önsöz

İpuçlarının farklı bir derlemesi mi ? Evet! Eğer ilk adminlik kursunuzu tamamlamışsanız ya da işlerinizi halletmek için en uygun yollara bakıyorsanız “Linux'ta 101 ipucu” adlı e-kitap başlamak için iyi bir nokta olacaktır. Bu kullanışlı ipuçları kolay okumanız ve anlamanız için yazıldı.

Tebrik ederim! Bu e-kitabı öğrencilerime önereceğim.

**- Prof. Dr. Fritz Mehner , FH Südwestfalen , Almanya
(bash-support vim eklentisi de dahil birkaç vim eklentisinin yazarı)**

Versiyon

Versiyon	Tarih	Baskı
1.0	12 Şubat 2009	İlk Düzenleme

Kitabın en son sürümünü [buradan](#) indirebilirsiniz. (İngilizce)

Bölüm 1: Güçlü CD Komutu İpuçları

cd, bir Unix oturumunda en sık kullanılan komuttur. Bu bölümde bahsedilen 6 adet, cd komutu hakkındaki ipucu anında sizin verimliliğinizi artıracak ve komut satırındaki dizin yapıları arasında gezinmenizi kolaylaştıracak.

İpucu 1. CDPATH kullanarak cd komutu için anadizin tanımlamak

Eğer belli bir ana dizinin alt dizinlerine ulaşmak için cd komutunu sıklıkla kullanıyorsanız ana dizini gösteren CDPATH'ı ayarlayabilirsiniz ve ana dizinin yolunu vermeden altdizinlere ulaşabilirsiniz, aşağıda açıklandığı gibi.

```
[ramesh@dev-db ~]# pwd
/home/ramesh

[ramesh@dev-db ~]# cd mail
-bash: cd: mail: Dosya veya dizin bulunamadı.

[Not: Bulunduğu dizinin içinde mail adında bir dizini arıyor.]

[ramesh@dev-db ~]# export CDPATH=/etc
[ramesh@dev-db ~]# cd mail
/etc/mail

[Not: /etc dizininin altında mail adında bir dizini arıyor.]

[ramesh@dev-db /etc/mail]# pwd
/etc/mail
```

Bu ayarları kalıcı olarak değiştirmek isterseniz, ~/.bash_profile dosyanıza “export CDPATH=/etc” yi ekleyebilirsiniz.

PATH değişkenine benzer olarak CDPATH değişkende “:” ile ayırarak birden fazla dizin girdisi ekleyebilirsiniz.

```
export CDPATH=.:~/etc:/var
```

Bu ipucu, aşağıdaki durumlarda size çok yardımcı olacaktır:

- o Oracle DBA'lar sıklıkla \$ORACLE_HOME altında çalışır , CDPATH değişkenini oracle home'a ayarlayabilirsiniz.
- o Unix sistem yöneticileri sıklıkla /etc altında çalışır , CDPATH değişkenini /etc ye ayarlayabilirsiniz.
- o Programcılar sıklıkla proje dizini altında (/home/projects) çalışır , CDPATH değişkenini /home/projects'e ayarlayabilirsiniz.
- o Son kullanıcılar sıklıkla alt dizinlere , ana dizinleri altından ulaşırlar , CDPATH değişkenini ~ 'e (ana dizin) ayarlayabilirsiniz.

İpucu 2. Dizinlerde etkili şekilde gezinmek için cd alias'ı kullanma

Uzun dizin yapılarında önceki dizine gelmek için cd ../../ kullanmalısınız. Kaç dizin üste çıkmak istediğinize bağlı olarak ../../yi kullanmanın bir çeşit kısayolu aşağıda anlatılmıştır.

```
# mkdir -p
/tmp/very/long/directory/structure/that/is/too/deep
# cd /tmp/very/long/directory/structure/that/is/too/deep
# pwd
/tmp/very/long/directory/structure/that/is/too/deep
# cd ../../../../
# pwd
/tmp/very/long/directory/structure
```

Bulduğumuz dizinin dört seviye üstüne çıkmak için cd ../../../../ komutu yerine aşağıdaki üç alias (takma ad) yönteminden birini kullanın.

Yöntem 1: “..n” kullanarak üst dizine çıkmak

Aşağıdaki örnekte, ..4 komutu 4 dizin (yerine göre seviye de denilebilir) yukarı çıkmak için , kısaca ..n komutu n dizin yukarı çıkmak için kullanılmıştır.Aşağıdaki alias (takma ad)'ları ~/.bash-profile dosyanıza ekleyin ve yeniden giriş yapın.

```
alias ..="cd .."  
alias ..2="cd ../../"  
alias ..3="cd ../../.."  
alias ..4="cd ../../../../"  
alias ..5="cd ../../../../.."
```

```
# cd /tmp/very/long/directory/structure/that/is/too/deep  
# ..4  
[Not: ..4'ü 4 seviye yukarı çıkmak için kullandık.]  
# pwd  
/tmp/very/long/directory/structure/
```

Yöntem 2: Sadece noktalar kullanarak üst dizine çıkmak

Aşağıdaki örnekte, (beş nokta) 4 dizin yukarı çıkmak için kullanıldı. Beş nokta yazarak 4 dizin üste çıkma hatırlama açısından gerçekten kolay, ilk iki noktayı yazdığınız zaman , bir dizin üste çıkacağını söylüyorsunuz , sonraki eklediğiniz her nokta bir dizin üste çıkmanızı sağlıyor. Uygun şekilde çalışması için ~/.bash_profile dosyanıza aşağıdaki kodları ekleyin ve yeniden giriş yapın.

```
alias ..="cd .."  
alias ...="cd ../../"  
alias ....="cd ../../.."  
alias .....="cd ../../../../"  
alias .....="cd ../../../../.."  
  
# cd /tmp/very/long/directory/structure/that/is/too/deep  
# .....  
  
[Not: ..... (beş nokta)'yı 4 dizin yukarı çıkmak için kullandık.]  
  
# pwd  
/tmp/very/long/directory/structure/
```

Yöntem 3: cd'yi tâkiben eklenen numaralarla üst dizine çıkmak

Aşağıdaki örnekte, cd4 (cd'den sonraki 4 rakamı) dört dizin üste çıkmak için kullanılmıştır.

```
alias cd1="cd .."  
alias cd2="cd ../../"  
alias cd3="cd ../../.."  
alias cd4="cd ../../../../"
```

İpucu 3. mkdir ve cd'yi tek bir komutta uygulamak

Bazen yeni bir dizin oluşturduğunuzda, hemen aşağıdaki gibi cd uygulayabilirsiniz.

```
# mkdir -p /tmp/subdir1/subdir2/subdir3  
  
# cd /tmp/subdir1/subdir2/subdir3  
  
# pwd  
/tmp/subdir1/subdir2/subdir3
```

mkdir ve cd komutunu tek bir satırda birleştirmek iyi olmaz mıydı ? Aşağıdaki satırı .bash-profile dosyanıza ekleyin ve yeniden giriş yapın.

```
$ vi .bash_profile  
function mkdircd () { mkdir -p "$@" && eval cd "\"\${$#}\""; }
```

Şimdi, mkdir ve cd'yi tek bir komutla aynı satırda çalıştırabiliriz. Aşağıdaki örneği inceleyiniz.

```
# mkdircd /tmp/subdir1/subdir2/subdir3  
  
[Not: Dizini oluşturuyor ve otomatik olarak cd komutunu kullanıyor.]  
  
# pwd  
/tmp/subdir1/subdir2/subdir3
```

İpucu 4. Son iki dizin arasında geçiş yapmak için “cd -” kullanmak

Aşağıda gösterildiği gibi siz de en son gezdiğiniz iki dizin arasında “cd -” kullanarak geçiş yapabilirsiniz.

```
# cd /tmp/very/long/directory/structure/that/is/too/deep  
  
# cd /tmp/subdir1/subdir2/subdir3  
  
# cd -
```

```
# pwd
/tmp/very/long/directory/structure/that/is/too/deep

# cd -

# pwd
/tmp/subdir1/subdir2/subdir3

# cd -

# pwd
/tmp/very/long/directory/structure/that/is/too/deep
```

İpucu 5. dirs , pushd ve popd kullanarak dizin yığınlarını idare etme

Dizinlerinizi içine atmak için dizin yığınlarını kullanabilirsiniz. Bu örnekte kullanılan üç komutu inceleyin.

- o dirs: Dizin yığını gösterir.
- o pushd: Dizini, yığının içine iter.
- o popd: Yığından, dizini çeker(en üsttekini alır) ve cd ile dizine girer.

Dirs her zaman yığının içeriğindeki geçerli dizini bastıracaktır. Üstelik dizin yığını boş olduğunda dirs komutu hâlâ aşağıdaki gibi sadece geçerli dizini bastıracaktır.

```
# popd
-bash: popd: dizin yığını boş.
# dirs
~
# pwd
/home/ramesh
```

Pushd ve popd'yi nasıl kullanırız? Hadi ilk geçici dizinlerimizi oluşturalım ve aşağıdaki gibi onları dizin yığına itelim.

```
# mkdir /tmp/dir1

# mkdir /tmp/dir2
# mkdir /tmp/dir3
# mkdir /tmp/dir4

# cd /tmp/dir1
# pushd .
```

```
# cd /tmp/dir2
# pushd .
# cd /tmp/dir3
# pushd .
# cd /tmp/dir4
# pushd .

# dirs

/tmp/dir4 /tmp/dir4 /tmp/dir3 /tmp/dir2 /tmp/dir1

[Not: İlk dizinin (/tmp/dir4) dir komutu çıktısı daima geçerli dizin
olacaktır yığının içeriğinden değil.]
```

Bu aşamada, dizin yığını aşağıdaki dizinleri içerir:

```
/tmp/dir4
/tmp/dir3
/tmp/dir2
/tmp/dir1
```

Yığına ittiğiniz son dizin en üstte olacaktır. Popd uyguladığınız zaman, yığında en üstteki dizine cd uygular ve onu yığından kaldırır. Yukarıda gösterildiği şekilde, yığının içine atılan son dizin /tmp/dir4'dir. Bu yüzden, aşağıdaki gibi biz popd işlemi yaparken, cd ile /tmp/dir4'e girecek ve onu yığından kaldıracak.

```
# popd
# pwd
/tmp/dir4
[Not: Yukarıdaki popd'dan sonra, dizin yığını içeriği şöyle olacaktır:
/tmp/dir3
/tmp/dir2
/tmp/dir1]
# popd
# pwd
/tmp/dir3
[Not: Bu popd'dan sonra, dizin yığını içeriği şöyle olacaktır:
/tmp/dir2
/tmp/dir1]
# popd

# pwd

/tmp/dir2
[Not: Bu popd'dan sonra, dizin yığını içeriği: /tmp/dir1]
```

```
# popd
# pwd
/tmp/dir1
[Not: Yukarıdaki popd'dan sonra, dizin yığını boş olacaktır!]
# popd
-bash: popd: dizin yığını boş.
```

İpucu 6. “shopt -s cdspell” kullanarak cd’de yanlış yazılmış dizin adını otomatik olarak düzeltme

shopt -s cdspell kullanarak aşağıda gösterildiği gibi cd komutundaki yazım hatalarını otomatik olarak düzeltebilirsiniz. Eğer klavyeyle aranız pek iyi değilse çok hata yaparsınız, bu çok yararlı olacaktır.

```
# cd /etc/mall
-bash: cd: /etc/mall: Dosya veya dizin bulunamadı.

# shopt -s cdspell

# cd /etc/mall
# pwd
/etc/mail

[Not: Yanlışlıkla, mail yerine mall yazdım, cd otomatik olarak düzeltti.]
```

Bölüm 2: Tarih Yönetimi

İpucu 7. Sistemin tarih ve saatini ayarlama

Sistem tarihini değiştirmek için şunu kullanırız:

```
# date {aagssdkyyyy.ss}

o aa - Ay
o gg - Gün
o ss - 24 saat formatında
o dk - Dakika
o yyyy - Yıl
o ss - Saniye
```

Örneğin sistem tarihini 31 Ocak 2009, öğleden sonra 10.19 ve 53. saniyesi için ayarlayalım.

```
# date 013122192009.53
```

Ayrıca aşağıda gösterildiği gibi argümanlar kullanarak da sistem tarihini değiştirebilirsiniz.

```
# date 013122192009.53
# date +%Y%m%d -s "20090131"
# date -s "01/31/2009 22:19:53"
# date -s "31 JAN 2009 22:19:53"
# date --set="31 JAN 2009 22:19:53"
```

Sadece saati düzenlemek için:

```
# date +%T -s "22:19:53"
# date +%T%p -s "10:19:53PM"
```

İpucu 8. Donanımın tarih ve saatini ayarlama

Donanımın tarih ve saatini ayarlamadan önce, işletim sisteminin tarih ve saatinin ipucu 7'ye uygun bir şekilde ayarlandığından emin olun. Donanımın tarih ve saatini sistem tarihini esas alarak aşağıdaki gibi ayarlayabilirsiniz.

```
# hwclock --systohc
# hwclock --systohc -utc
```

Hiçbir parametre olmadan hwclock komutu ile geçerli donanım tarih ve saatini gösterebiliriz.

```
# hwclock
```

Sistemin UTC(eşgüdümlü evrensel zaman) için ayarlanmış olup olmadığını doğrulamak için saat dosyasını şöyle denetleyebilirsiniz:

```
# cat /etc/sysconfig/clock
ZONE="America/Los_Angeles"
UTC=false
ARC=false
```


İpucu 9. Şuanki tarihi ve saati özel formatta gösterme

Aşağıda çeşitli formatlarda şimdiki tarih ve saati yazdırmanın farklı yolları gösterilmiştir:

```
$ date
Prş Oca 1 08:19:23 PST 2009

$ date --date="now"
Prş Oca 1 08:20:05 PST 2009

$ date --date="today"
Prş Oca 1 08:20:12 PST 2009

$ date --date='1970-01-01 00:00:01 UTC +5 hours' +%s
18001

$ date '+Şuan tarih: %m/%d/%y%nŞuan saat:%H:%M:%S'
Şuan tarih: 01/01/09
Şuan saat:08:21:41

$ date +"%d-%m-%Y"
01-01-2009

$ date +"%d/%m/%Y"
01/01/2009

$ date +"%A,%B %d %Y"
Perşembe,Ocak 01 2009
```

Aşağıda date komutu için yazabileceğiniz farklı format seçenekleri bulunuyor:

- o %D tarih (aa/gg/yy)
- o %d ayın günü (01..31)
- o %m ay (01..12)
- o %y yılın son iki rakamı (00..99)
- o %a haftanın günlerinin kısaltılmış hâli (Paz...Cmt)
- o %A haftanın günlerinin tam hâli (Pazar...Cumartesi)
- o %b ayların kısaltılmış hâli (Oca..Ara)
- o %B ayların tam hâli (Ocak..Aralık)
- o %H saat (00..23)
- o %I saat (01..12)
- o %Y yıl (1970...)

İpucu 10. Geçmiş bir tarihi ve saati gösterme

Aşağıda geçmiş tarih ve saati göstermenin çeşitli yönü gösterilmiştir:

```
$ date --date='3 seconds ago'  
Prş Oca 1 08:27:00 PST 2009  
  
$ date --date="1 day ago"  
Çrş Ara 31 08:27:13 PST 2008  
  
$ date --date="1 days ago"  
Çrş Ara 31 08:27:18 PST 2008  
  
$ date --date="1 month ago"  
Pzt Ara 1 08:27:23 PST 2008  
  
$ date --date="1 year ago"  
Sal Oca 1 08:27:28 PST 2008  
  
$ date --date="yesterday"  
Çrş Ara 31 08:27:34 PST 2008  
  
$ date --date="10 months 2 day ago"  
Prş Şub 28 08:27:41 PST 2008
```

İpucu 11. Gelecek bir tarihi ve saati gösterme

Aşağıdaki örneklerde nasıl gelecek tarihi ve saati göstereceğiniz gösterilmiştir:

```
$ date  
Prş Oca 1 08:30:07 PST 2009  
  
$ date --date='3 seconds'  
Prş Oca 1 08:30:12 PST 2009  
  
$ date --date='4 hours'  
Prş Oca 1 12:30:17 PST 2009  
  
$ date --date='tomorrow'  
Cum Oca 2 08:30:25 PST 2009  
  
$ date --date="1 day"  
Cum Oca 2 08:30:31 PST 2009  
  
$ date --date="1 days"  
Cum Oca 2 08:30:38 PST 2009  
  
$ date --date="2 days"  
Cmt Oca 3 08:30:43 PST 2009
```

```
$ date --date='1 month'
Paz Şub 1 08:30:48 PST 2009

$ date --date='1 week'
Prş Oca 8 08:30:53 PST 2009

$ date --date="2 months"
Paz Mar 1 08:30:58 PST 2009

$ date --date="2 years"
Cmt Oca 1 08:31:03 PST 2011

$ date --date="next day"
Cum Oca 2 08:31:10 PST 2009

$ date --date="-1 days ago"
Cum Oca 2 08:31:15 PST 2009

$ date --date="this Wednesday"
Çrş Oca 7 00:00:00 PST 2009
```

Bölüm 3: SSH İstemci Komutları

İpucu 12. SSH istemci versiyonunu öğrenmek

Bazen hâlihazırda çalışan ve aralarında bağ kurulmuş olan SSH istemcisinin versiyon numarasını öğrenmeniz gerekebilir. “ssh -V” kullanarak versiyon numarasını öğrenebilirsiniz.Şunu unutmayın, Linux OpenSSH ile gelir. Sıradaki örnek bu özel sistemin OpenSSH kullandığını belirtir:

```
$ ssh -V
OpenSSH_3.9p1, OpenSSL 0.9.7a Feb 19 2003
```

Aşağıdaki örnek bu özel sistemin SSH2 kullandığını belirtir:

```
$ ssh -V
ssh: SSH Secure Shell 3.2.9.1 (non-commercial version) on i686-pc-linux-gnu
```

İpucu 13. SSH ile uzak sunucuda oturum açma

İlk defa bir yerel sunucudan(localhost) uzak sunucuya(remotehost) giriş yaptığınızda, “Host key bulunamadı” mesajı alırsınız ve devam etmek için “yes” yazmanız gerekir. Host key’i sunucu kimliği olarak çevirebiliriz.

Uzak sunucunun sunucu kimliği home dizininizin altında .ssh2/hostkeys dizini altına eklenir, aşağıda gösterildiği gibi.

```
localhost$ ssh -l jsmith remotehost.example.com

Veritabanında sunucu kimliği bulunamadı.
Anahtar parmakizi:
xabie-dezbc-manud-bartd-satsy-limit-nexiu-jambl-title-jarde-tuxum
Anahtar dosyasında % ssh-keygen -F ortkanahtar.pub çalıştırarak ortak anahtar
parmakizini alabilirsin.

Bağlantıya devam etmek istediğine emin misin(yes/no)? Yes
Sunucu kimliği /home/jsmith/.ssh2/hostkeys/key_22_remotehost.example.com.pub'e
kaydedildi.
remotehost.example.com için sunucu kimliği, jsmith tarafından Pzt May 26 2008
16:06:50 kabul edildi.
jsmith@remotehost.example.com parola:
remotehost.example.com$
```

İkinci defa yerel sunucudan uzak sunucuya giriş yaptığınızda, yalnızca parolayı soran bir prompt(girdi bekleyen satır) görürsünüz, uzak sunucu kimliği zaten ssh istemcisinin bilinen sunucular listesine eklenmiştir.

```
localhost$ ssh -l jsmith remotehost.example.com
jsmith@remotehost.example.com parola:

remotehost.example.com$
```

Bazı sebeplerden dolayı, eğer uzak sunucunun sunucu kimliği ilk girişten sonra değişmişse, aşağıdaki gibi bir uyarı mesajı alabilirsiniz. Bu durum, aşağıdaki gibi çeşitli nedenlerden dolayı olabilir:

- o Uzak sunucuda SSH sunucusu güncellenmiştir ya da yeniden yüklenmiştir.
- o Birileri kötü niyetli aktivite yapıyor olabilir vb.

Aşağıdaki mesaja “yes” demeden önce yapabileceğiniz mümkün olan en iyi durum, sistem yöneticinizi aramanız ve neden sunucu kimliğinin değiştiğini belirlemeniz ve sunucu kimliğinin doğru olup olmadığını kontrol etmeniz.

```
localhost$ ssh -l jsmith remotehost.example.com
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@  UYARI: SUNUCU KİMLİK BİLGİLERİ DEĞİŞTİ  @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
BİRİNİN KÖTÜ BİR ŞEY YAPMASI MÜMKÜN!
Birisi şu anda sizin verilerinizi dinleyebilir. (man-in-the- middle saldırısı)!
```

```
Ayrıca sunucu kimliği sadece değişmiş de olabilir.  
Lütfen sistem yöneticinizle iletişime geçiniz.  
Bu mesajdan kurtulmak için doğru sunucu kimliğini  
"/home/jsmith/.ssh2/hostkeys/key_22_remotehost.example.com.pub" dosyasına ekleyin.  
  
Geçerli sunucu kimliğinin parmakizi:  
xabie-dezbc-manud-bartd-satsy-limit-nexiu-jambl-title-arde-tuxum  
Anahtar dosyasında % ssh-keygen -F ortakanahtar.pub çalıştırarak ortak anahtar  
parmakizini alabilirsiniz.  
Temsilci yönlendirme bozuk sunucular tarafından saldırıları önlemek için devre dışı  
bırakılır.  
Bağlantıya devam etmek istediğinizden emin misiniz(yes/no)? yes  
Diskteki sunucu kimliğini değiştirmek ister misiniz(yes/no)? yes  
  
Temsilci yönlendirme tekrar aktif.  
Sunucu kimliği /home/jsmith/.ssh2/hostkeys/key_22_remotehost.example.com.pub  
kaydedildi.  
remotehost.example.com için sunucu kimliği, jsmith tarafından Pzt May 26 2008 16:17:31  
kabul edildi.  
jsmith@remotehost.example.com parola:  
remotehost$
```

İpucu 14. SSH istemci oturumunda hata ayıklama

Bazen SSH bağlantı sorunları için hata ayıklama mesajlarını görmemiz gerekebilir. -v (küçük v) seçeneği ile aşağıdaki gibi SSH hata ayıklama mesajlarını yazdırabilirsiniz.

SSH istemci hata ayıklama mesajları olmadan:

```
localhost$ ssh -l jsmith remotehost.example.com  
  
uyarı: remotehost.example.com bağlantısı başarısız oldu: Adı ile ilişkili  
adres bulunamadı.
```

SSH istemci hata ayıklama mesajları ile:

```
localhost$ ssh -v -l jsmith remotehost.example.com  
  
debug: SshConfig/sshconfig.c:2838/ssh2_parse_config_ext: Metaconfig ayrıştırma  
satır 3'te durdu.  
  
debug: SshConfig/sshconfig.c:637/ssh_config_set_param_verbose: 'VerboseMode'  
değişkeni 'FALSE' olarak ayarlandı.  
  
debug: SshConfig/sshconfig.c:3130/ssh_config_read_file_ext: Config dosyasından  
17 parametre okundu.  
  
debug: Ssh2/ssh2.c:1707/main: Kullanıcı config dosyası bulunamadı,  
varsayılan kullanılacak. ('/home/jsmith/.ssh2/ssh2_config' dosyasına bakın.)
```

```
debug: remotehost.example.com'e bağlanma , port 22... (SOCKS kullanılmadı)
uyarı: remotehost.example.com'e bağlanma başarısız oldu: Adı ile ilişkili
adres bulunamadı.
```

İpucu 15. SSH kaçış karakterleri ile SSH oturumunda geçiş yapmak

SSH kullanarak yerel sunucudan uzak sunucuda oturum açtığınızda, birkaç aktivite yapmak ve uzak sunucuya tekrar dönmek için yerel sunucuya tekrar dönmek isteyebilirsiniz. Bu durumda, uzak sunucudaki ssh oturumunuzu sonlandırmanız gerekmez. Bunun yerine, aşağıdaki adımları izleyebilirsiniz.

1. Yerel sunucudan uzak sunucuda oturum açma:

```
localhost$ ssh -l jsmith remotehost
```

2. Şimdi uzak sunucuya bağlandık:

```
remotehost$
```

3. Geçici olarak yerel sunucuya dönmek istersek, ~ (alt gr+ü) kaçış karakterine basın ve Ctrl+Z yapın.

~ yazdığınızda Ctrl+Z yapıp enter'a basmadıkça ekranda hemen bir şey görmezsiniz. Bu yüzden, işe yaraması için uzak sunucuda şöyle girin: ~<Ctrl+Z>

```
remotehost$ ~^Z
[1]+ ssh -l jsmith remotehost durduruldu.
```

```
localhost$
```

4. Şimdi yerel sunucuya geri döndünüz ve SSH uzak sunucu oturumu UNIX arkaplanında işlem olarak çalışır, aşağıdaki gibi kontrol edebilirsiniz:

```
localhost$ jobs
[1]+ ssh -l jsmith remotehost durduruldu.
```

5. Yerel sunucuda arkaplandaki uzak sunucu oturumunu önplana getirerek tekrar şifre girmeden uzak sunucuya bağlanabilirsiniz.

```
localhost$ fg %1
ssh -l jsmith remotehost
remotehost$
```

İpucu 16. SSH kaçış karakterleri ile SSH oturum istatistikleri

Geçerli SSH oturumu hakkında kullanışlı istatistikleri, aşağıdaki gibi alabiliriz. Bu işlem sadece SSH2 istemcisinde çalışır.

1. Yerel sunucudan uzak sunucuda oturum açın.

```
localhost$ ssh -l jsmith remotehost
```

2. Uzak sunucuda, aşağıdaki gibi “~” SSH kaçış karakteri sonrasında s yapabilirsiniz. Bu, geçerli SSH bağlantınız hakkında çokça kullanışlı istatistik gösterir.

```
remotehost$ [Not: ~s komut satırında yazarken herhangi bir şey gözükmez.]
```

```
uzak sunucu: remotehost
yerel sunucu: localhost
uzak bağlantı versiyon: SSH-1.99-OpenSSH_3.9p1
yerel versiyon: SSH-2.0-3.2.9.1 SSH Secure
Kabuk (Ticari olmayan)
gelen sıkıştırılmış bayt: 1506
gelen sıkıştırılmamış bayt: 1622
giden sıkıştırılmış bayt: 4997
giden sıkıştırılmamış bayt: 5118
gelen paket: 15
giden paket: 24
tekrar girdi: 0
Algoritmalar:
Seçilen anahtar değişim algoritması: diffie-hellman-group1-sha1
Seçilen sunucu anahtar algoritması: ssh-dss
Yaygın sunucu anahtar algoritmaları: ssh-dss,ssh-rsa
İstemciden sunucuya algoritmalar:
Şifreleme: aes128-cbc
MAC: hmac-sha1
Sıkıştırma: zlib

Sunucudan istemciye algoritmalar:
Şifreleme: aes128-cbc
MAC: hmac-sha1
Sıkıştırma: zlib
```

```
localhost$
```

Bir dipnot, SSH anahtar tabanlı kimlik doğrulama yüklediyseniz, OpenSSH ve SSH2 dökümanlarını inceleyebilirsiniz.

Bölüm 4: Başlıca Linux Komutları

İpucu 17. Grep komutu

grep komutu belirli bir metin içerisinde istenen kelimeleri vb. aramak için kullanılır. Bu inanılmaz güçlü komut, birkaç seçenekle birlikte kullanılır.

Söz dizimi : grep [seçenek] aranacak_kısım [dosya yolu]

Bir dosya içindeki belirli kelimelerle eşleşen satırları nasıl bulabilirim ?

Bu örnekte, grep, /etc/passwd dosyası içinde John kelimesine bakar ve tüm eşleşen satırları gösterir.

```
# grep John /etc/passwd  
  
jsmith:x:1082:1082:John Smith:/home/jsmith:/bin/bash  
jdoe:x:1083:1083:John Doe:/home/jdoe:/bin/bash
```

-v seçeneği, eşleşen satırlar dışındaki tüm satırları gösterir. Aşağıdaki örnekte, /etc/passwd 'deki John ile uymayan tüm kayıtları gösterir.

```
# grep -v John /etc/passwd  
  
jbourne:x:1084:1084:Jason Bourne:/home/jbourne:/bin/bash
```

Belirli dosyadaki metinde eşleşen kaç satır var?

Aşağıdaki örnekte, /etc/passwd dosyasındaki metinde John içeren toplam satır sayısı gösterilmiştir.

```
# grep -c John /etc/passwd  
2
```

Ayrıca -cv parametresi ile belirtilen kelime ile eşleşmeyen toplam satır sayısını da yazdırabilirsiniz.


```
# grep -cv John /etc/passwd
39
```

Küçük büyük farketmeksizin bir metin nasıl aranır ?

-i (ignore case) seçeneği ile, aranan nesnenin küçük büyük harf olup olmaması dikkate alınmaksızın aranır.

```
# grep -i john /etc/passwd

jsmith:x:1082:1082:John Smith:/home/jsmith:/bin/bash
jdoe:x:1083:1083:John Doe:/home/jdoe:/bin/bash
```

Metinde belirtilen kelimeleri tüm altdizinlerde nasıl arayabilirim ?

-r (recursive=>özyinelemeli) parametresi ile bunu yapabiliriz. Aşağıdaki örnekte /home/users altındaki tüm altdizinlerin içerisinde John kelimesini küçük büyük harf gözetmeksizin arayacaktır.

Bu komut “dosya adı:eşleşen satır” formatında bir çıktı gösterecektir. Ayrıca -l seçeneği ile sadece eşleşen dosya adını da yazdırabilirsiniz.

```
# grep -ri john /home/users

/home/users/subdir1/letter.txt:John, Thanks for your contribution.
/home/users/name_list.txt:John Smith
/home/users/name_list.txt:John Doe

# grep -ril john /root

/home/users/subdir1/letter.txt
/home/users/name_list.txt
```

İpucu 18. Find komutu

find komutu, UNIX dosya sisteminde çok sayıda koşula göre dosya bulmak için sıklıkla kullanılan komuttur. Hadi bir örnekle find komutunu inceleyelim.

```
Söz dizimi: find [dosya yolu] [durumlar]
```

Adında bulunan kelimeyi içeren dosyalar nasıl bulunur ?

Aşağıdaki komut /etc dizini altında dosya adında mail bulunan tüm dosyalara bakıyor.

```
# find /etc -name "*mail*"
```

Belirlenmiş boyuttan daha büyük tüm dosyalar nasıl bulunur ?

Aşağıdaki komut sistemde 100 MB'dan büyük olan tüm dosyaları listeler.

```
# find / -type f -size +100M
```

Son x gündür düzenlenmeyen dosyalar nasıl bulunur ?

Aşağıdaki komut geçerli dizin altında 60 günden daha uzun süredir düzenlenmeyen dosyaları listeler. (60+ gündür düzenlenmemiş)

```
# find . -mtime +60
```

Son x günde düzenlenmiş dosyalar nasıl bulunur ?

Aşağıdaki komut geçerli dizin altında son 2 günde düzenlenmiş olan dosyaları listeler.

```
# find . -mtime -2
```

*tar.gz uzantılı ve 100MB'dan büyük arşiv dosyaları nasıl silinir ?

Yanlışlıkla dosyaları silmek istemiyorsanız aşağıdaki komutları çalıştırırken lütfen dikkatli olun. rm ile komutu çalıştırdığınızda hangi dosyanın silineceğine emin olmak için en iyi uygulama, ls -l ile çalıştırmaktır.

```
# find / -type f -name *.tar.gz -size +100M -exec ls -l {} \;  
# find / -type f -name *.tar.gz -size +100M -exec rm -f {} \;
```

Son x gündür düzenlenmemiş dosyalar nasıl arşivlenir ?

Aşağıdaki komut /home/jsmith altındaki son 60 gün içinde düzenlenmemiş tüm dosyaları bulur ve /tmp altında ggaayyyy_archive.tar formatında yeni bir arşiv oluşturur.

```
# find /home/jsmith -type f -mtime +60 | xargs tar -cvf
/tmp/`date '+%d%m%Y'`_archive.tar`
```

Dipnot, güçlü bir metin tabanlı dosya yönetim programı olan Midnight Commander GUI kullanarak dosya ile ilgili birçok şey gerçekleştirebilirsiniz. (dosyaları bulma dâhil)

İpucu 19. Standart çıktı ve hata mesajlarını yönlendirmek

Bazen bir kabuk betiğinde hata ayıklarken, standart çıktı veya hata mesajlarını görmek istemeyebilirsiniz. Çıktıyı yönlendirmek için aşağıdaki gibi /dev/null kullanabilirsiniz.

> kullanarak standart çıktıyı /dev/null 'a yönlendirme

echo ifadesi ile göstermek istemediğinizde ve sadece hata mesajlarına bakmakla ilgilendiğinizde bu işlem çok yardımcı olur.

```
# cat file.txt > /dev/null
# ./shell-script.sh > /dev/null
```

2> kullanarak standart hataları /dev/null 'a yönlendirme

Sadece standart çıktıyı göstermekle ilgileniyorsanız ve hata mesajlarını göstermek istemiyorsanız, bu işlem yardımcı olacaktır.

```
# cat invalid-file-name.txt 2> /dev/null
# ./shell-script.sh 2> /dev/null
```

İpucu 20. Join komutu

Join komutu iki dosyadaki satırları ortak bir alana göre birleştirir. Aşağıdaki örnekte, işçi.txt ve maaş.txt adında iki dosyamız var. İkisi de işçi numarası adında ortak bir alana sahip. Join komutunu aşağıdaki gibi iki dosyadaki işçi numaralarından birleştirme yapmak için kullanabiliriz.

```
$ cat işçi.txt
100 Jason Smith
200 John Doe
300 Sanjay Gupta
400 Ashok Sharma
```

```
$ cat maaş.txt
100 $5,000
200 $500
300 $3,000
400 $1,250

$ join işçi.txt maaş.txt
100 Jason Smith $5,000
200 John Doe $500
300 Sanjay Gupta $3,000
400 Ashok Sharma $1,250
```

İpucu 21. Harf değiştirme

Dosyadaki her şeyi büyük harfe dönüştürme

```
$ cat işçi.txt
100 Jason Smith
200 John Doe
300 Sanjay Gupta
400 Ashok Sharma

$ tr a-z A-Z < işçi.txt
100 JASON SMITH
200 JOHN DOE
300 SANJAY GUPTA
400 ASHOK SHARMA
```

Dosyadaki her şeyi küçük harfe dönüştürme

```
$ cat bölüm.txt
100 FINANCE
200 MARKETING
300 PRODUCT DEVELOPMENT
400 SALES

$ tr A-Z a-z < bölüm.txt
100 finance
200 marketing
300 product development
400 sales
```

İpucu 22. Xargs komutu

xargs, komutun çıktısını alan ve onu başka bir komuta argüman olarak veren çok güçlü bir komuttur. Aşağıda, xargs komutunu nasıl etkili biçimde kullanabileceğinize dair birkaç örnek var.

1. rm kullanarak çok fazla dosyayı silmek istediğiniz zaman, /bin/rm Argüman listesi çok uzun hatası alırsınız. Bu problemi önlemek için xargs kullanabilirsiniz.

```
find ~ -name '*.log' -print0 | xargs -0 rm -f
```

2. /etc altındaki tüm *.conf dosyalarını listelerken kullanabilirsiniz. Aynı sonucu almanın farklı yolları da var. Aşağıdaki örnekte, sadece xargs kullanılarak gösteriliyor. Bu örnekteki find komutunun çıktısı, kullanılan xargs tarafından ls -l 'ye veriliyor.

```
# find /etc -name "*.conf" | xargs ls -l
```

3. Eğer indirmek istediğiniz dosyaların URL listesi olan bir dosyanız varsa, aşağıdaki gibi xargs kullanabilirsiniz.

```
# cat url-list.txt | xargs wget -c
```

4. Tüm jpg resimleri bulup arşiv yapabilirsiniz.

```
# find / -name *.jpg -type f -print | xargs tar -cvzf images.tar.gz
```

5. Tüm resimleri harici bir harddiske kopyalayabilirsiniz.

```
# ls *.jpg | xargs -n1 -i cp {} /harici-hard-disk/dizin
```

İpucu 23. Sort komutu

Sort komutu bir metin dosyasındaki satırları sıralar. Aşağıdaki birkaç örnekte “işçi_adı:işçi_numarası:bölüm_adı” formatında veriler içeren metin dosyalarında sort komutunu nasıl kullanacağımız gösterilmiştir.

```
$ cat isimler.txt  
Emma Thomas:100:Marketing  
Alex Jason:200:Sales  
Madison Randy:300:Product Development  
Sanjay Gupta:400:Support  
Nisha Singh:500:Sales
```

Metin dosyasını küçükten büyüğe sıralamak

```
$ sort isimler.txt
Alex Jason:200:Sales
Emma Thomas:100:Marketing
Madison Randy:300:Product Development
Nisha Singh:500:Sales
Sanjay Gupta:400:Support
```

Metin dosyasını büyükten küçüğe sıralamak

```
$ sort -r isimler.txt
Sanjay Gupta:400:Support
Nisha Singh:500:Sales
Madison Randy:300:Product Development
Emma Thomas:100:Marketing
Alex Jason:200:Sales
```

“:” ile sınırlandırılmış metin dosyasında 2. alana göre sıralamak

```
$ sort -t: -k 2 isimler.txt
Emma Thomas:100:Marketing
Alex Jason:200:Sales
Madison Randy:300:Product Development
Sanjay Gupta:400:Support
Nisha Singh:500:Sales
```

Bir sekme ayrılmış metin dosyalarında 3. alanı(bölüm_adı) sıralamak ve kopyaları ortadan kaldırmak

```
$ sort -t: -u -k 3 isimler.txt
Emma Thomas:100:Marketing
Madison Randy:300:Product Development
Alex Jason:200:Sales
Sanjay Gupta:400:Support
```

passwd dosyasını, 3. alana(rakamsal kullanıcı adı) göre sıralamak

```
$ sort -t: -k 3n /etc/passwd | more
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
```

/etc/hosts dosyasını ip-adresine göre sıralama

```
$ sort -t . -k 1,1n -k 2,2n -k 3,3n -k 4,4n /etc/hosts
127.0.0.1 localhost.localdomain localhost
192.168.100.101 dev-db.thegeekstuff.com dev-db
192.168.100.102 prod-db.thegeekstuff.com prod-db
192.168.101.20 dev-web.thegeekstuff.com dev-web
192.168.101.21 prod-web.thegeekstuff.com prod-web
```

sort'u diğer komutlarla birleştirmek

- o `ps -ef | sort` : Process(işlem) listesinin çıktısını sıralar.
- o `ls -al | sort +4n` : Dosyaları boyutlarına göre küçükten büyüğe sıralar. Örnekte 5 alan sıralanır ve en küçük dosya ilk önce gösterilir.
- o `ls -al | sort +4nr` : Dosyaları boyutlarına göre büyükten küçüğe sıralar.Örnekte 5 alan sıralanır ve en büyük dosya ilk önce gösterilir.

İpucu 24. Uniq komutu

Uniq komutu, daha çok sort komutu ile birleştirilerek kullanılır. uniq sadece sıralanmış dosyadan kopyaları siler.uniq'in çalışması için, tüm kopyalar bitişik satırda olmalıdır.Aşağıda birkaç örnek vardır.

1. İçinde kopyalar bulunan bir işçi dosyasına sahip olduğunuzda, kopyaları silmek için aşağıdakini uygulayabilirsiniz.

```
$ sort namesd.txt | uniq
$ sort -u namesd.txt
```

2. Eğer kaç satır kopya olduğunu öğrenmek isterseniz, aşağıdakini uygulayın. Aşağıdaki örnekte ilk alan bulunduğu satırda kaç adet kopya bulunduğunu belirtir. Bu yüzden, aşağıdaki örnekte namesd.txt'de Alex ve Emma ile başlayan iki tane satır bulundu.

```
$ sort namesd.txt | uniq -c
2 Alex Jason:200:Sales
2 Emma Thomas:100:Marketing
1 Madison Randy:300:Product Development
1 Nisha Singh:500:Sales
1 Sanjay Gupta:400:Support
```

3. Aşağıda, sadece kopyası olan girdileri gösterir.

```
$ sort namesd.txt | uniq -cd
2 Alex Jason:200:Sales
2 Emma Thomas:100:Marketing
```

İpucu 25. Cut komutu

Cut komutu, metin dosyasından özel sütunları veya diğer komut çıktılarını göstermek için kullanılabilir.

Aşağıda birkaç örnek gösterilmiştir.

Sütunlara ayrılmış dosyada ilk alanı(işçi adı) gösterme

```
$ cut -d: -f 1 isimler.txt
Emma Thomas
Alex Jason
Madison Randy
Sanjay Gupta
Nisha Singh
```

Sütunlara ayrılmış dosyada 1. ve 3. alanları gösterme

```
$ cut -d: -f 1,3 isimler.txt
Emma Thomas:Marketing
Alex Jason:Sales
Madison Randy:Product Development
Sanjay Gupta:Support
Nisha Singh:Sales
```


Dosyadaki her satırın sadece ilk 8 harfini gösterme

```
$ cut -c 1-8 isimler.txt
Emma Tho
Alex Jas
Madison
Sanjay G
Nisha Si
```

Çeşitli Cut komutu örnekleri

- o "cut -d : -f 1 /etc/passwd" Sistemdeki tüm kullanıcıların unix kullanıcı adlarını gösterir.
- o "free | tr -s ' ' | sed '/^Mem!/d' | cut -d ' ' -f2" Sistemdeki toplam kullanılabilir belleği gösterir.

İpucu 26. Stat komutu

Stat komutu, tek bir dosyanın ya da dosya sisteminin durum/özelliklerini kontrol etmek için kullanılır.

Bir dosya veya dizinin istatistiklerini gösterme

```
$ stat /etc/my.cnf
Dosya: `/etc/my.cnf'
Boyut: 346 Blok: 16 GÇ(Girdi-Çıktı) Blok: 4096 normal dosya
Cihaz: 801h/2049d Inode: 279856 Linkler: 1
İzinler: (0644/-rw-r--r--) Uid: ( 0/ root) Gid: ( 0/ root)
Erişim: 2009-01-01 02:58:30.000000000 -0800
Düzenleme: 2006-06-01 20:42:27.000000000 -0700
Değişme: 2007-02-02 14:17:27.000000000 -0800

$ stat /home/ramesh
Dosya: `/home/ramesh'
Boyut: 4096 Blok: 8 GÇ Blok: 4096 dizin
Cihaz: 803h/2051d Inode: 5521409 Linkler: 7
İzinler: (0755/drwxr-xr-x) Uid: ( 401/ramesh) Gid: ( 401/ramesh)
Erişim: 2009-01-01 12:17:42.000000000 -0800
Düzenleme: 2009-01-01 12:07:33.000000000 -0800
Değişme: 2009-01-09 12:07:33.000000000 -0800
```

-f parametresini kullanarak dosya sisteminin durumunu gösterme

```
$ stat -f /
Dosya: "/"
ID: 0 Namelen: 255 Tip: ext2/ext3
Blok: Toplam: 2579457 Boş: 2008027 Uygun: 1876998 Boyut: 4096
Inode: Toplam: 1310720 Boş: 1215892
```

İpucu 27. Diff komutu

Diff komutu, iki farklı dosyayı karşılaştırır ve farklılıkları raporlar. Çıktısı şifreli-dir ve düz olarak okunamaz.

```
Söz dizimi: diff [seçenekler] dosya1 dosya2
```

Eski dosyayla karşılaştırıldığında yeni dosyada ne düzenlendi ?

Diff komutunda -w seçeneği karşılaştırma yaparken whitespace karakterleri(yazılı bir belgede görünmeyen karakterler) görmezden gelir.

Aşağıda diff çıktısı:

- o Satırın üstündeki --- ifadesi diff komutundaki ilk dosyada değişiklik olduğunu belirtir.(örnek isim_listesi.txt)
- o Satırın altındaki --- ifadesi diff komutundaki ikinci dosyada değişiklik olduğunu belirtir.(örnek yeni_isim_listesi.txt) İlk dosyaya ait olan satırlar < ile , ikinci dosyaya ait olan satırlar ise > ile başlar.

```
# diff -w isim_listesi.txt yeni_isim_listesi.txt
2c2,3
< John Doe
---
> John M Doe
> Jason Bourne
```

İpucu 28. Kullanıcıların toplam bağlı kalma sürelerini gösterme

ac komutu kullanıcıların bağlı kalma süreleri hakkındaki istatistikleri gösterir.

Geçerli bağlantısı bulunan kullanıcıların bağlantı süreleri

-d seçeneği, her günün çıktılarını ayrı ayrı verecektir. Aşağıdaki örnekte, ben bugün 6 saatten uzun süredir giriş yapmış bulunmaktayım. 1 Aralıkta ise, aşağı yukarı 1 saat bağlı kalmışım.

```
$ ac -d
Dec 1 total 1.08
Dec 2 total 0.99
Dec 3 total 3.39
Dec 4 total 4.50
Today total 6.10
```

Tüm kullanıcıların bağlı kalma süreleri

Tüm kullanıcıların bağlı kalma sürelerini göstermek için aşağıdaki gibi -p parametresini kullanmalısınız. Lütfen şunu unutmayın, bu bireysel kullanıcıların toplam bağlı kalma sürelerini belirtir.

```
$ ac -p
john          3.64
madison      0.06
sanjay       88.17
nisha       105.92
ramesh      111.42
total       309.21
```

Belirli bir kullanıcının bağlı kalma süresi

Belirli bir kullanıcının bağlı kalma süresini aşağıdaki komutu uygulayarak gösterebilirsiniz.

```
$ ac -d sanjay
Jul 2 total 12.85
Aug 25 total 5.05
Sep 3 total 1.03
Sep 4 total 5.37
Dec 24 total 8.15
Dec 29 total 1.42
Today total 2.95
```

Bölüm 5: PS1, PS2, PS3, PS4 ve PROMPT_COMMAND

İpucu 29. PS1 - Varsayılan Komut İstemcisi

Linuxtaki varsayılan etkileşimli komut istemcisini bilgilendirici ve kullanışlı olması için aşağıdaki gibi düzenleyebilirsiniz. Aşağıdaki örnekte, PS1 varsayılan olarak “\s-\v\\$” yani kabuk adını ve versiyon numarasını gösterir. Hadi aşağıdaki gibi varsayılan davranışını kullanıcı adını, bilgisayar adını ve geçerli dizinin adını göstermesi için değiştirelim.

```
-bash-3.2$ export PS1="\u@\h \w> "  
ramesh@dev-db ~> cd /etc/mail  
ramesh@dev-db /etc/mail>  
  
[Not:Komut istemcisi "kullanıcı_adi@bilgisayar_adi geçerli_dizin>" formatına göre değişti.]
```

Bu örnekte aşağıdaki PS1 kodları kullanıldı:

- o \u - Kullanıcı Adı (username)
- o \h - Bilgisayar Adı (hostname)
- o \w - Geçerli dizinin yol adresi. Unutmayın, home dizininde olduğunuzda yukarıda olduğu gibi sadece “~” gösterecektir.

Şunu unutmayın, PS1 değerlerinin sonunda bir boşluk karakteri var. Şahsen, ben kolay okunabilme açısından komut istemcisinin sonunda boşluk bırakmayı tercih ederim.

Aşağıda gösterildiği gibi ya `.bash_profile` ya da `.bashrc` dosyasının içeriğine `export PS1="\u@\h \w>` eklerseniz kalıcı olarak ayarlamış olursunuz.

```
ramesh@dev-db ~> vi ~/.bash_profile  
ramesh@dev-db ~> vi ~/.bashrc  
[Not: Yukarıdaki dosyaya export PS1="\u@\h \w> " satırını ekleyin.]
```

PS1'in detaylı kullanımı için birkaç pratik çözüm içeren sonraki bölüme bakın.

İpucu 30. PS2 – İkincil komut istemcisi

Uzun bir komutu birden fazla satırda yazmak için satırın sonuna \ ekleyerek yazabilirsiniz. Çoklu komutlar için varsayılan devam etkileşimli komut istemcisi ">" olur. Hadi aşağıdaki gibi PS2 çevre değişkeninin davranışını "continue->" gösterecek şekilde değiştirelim.

```
ramesh@dev-db ~> myisamchk --silent --force --fast --update-state \  
> --key_buffer_size=512M --sort_buffer_size=512M \  
> --read_buffer_size=4M --write_buffer_size=4M \  
> /var/lib/mysql/bugs/*.MYI
```

[Not: Devam eden satır için varsayılan olarak ">" kullanıyor.]

```
ramesh@dev-db ~> export PS2="continue-> "  
ramesh@dev-db ~> myisamchk --silent --force --fast --update-state \  
continue-> --key_buffer_size=512M --sort_buffer_size=512M \  
continue-> --read_buffer_size=4M --write_buffer_size=4M \  
continue-> /var/lib/mysql/bugs/*.MYI
```

[Not: Devam eden satır için "continue->" kullanıyor.]

\ kullanarak uzun komutları çoklu satırlara bölmeyi daha kullanışlı ve kolay okunaklı buluyorum. Ayrıca diğerlerine de baktım, kim uzun satırları bölmek istemez ki.

İpucu 31. PS3 - Kabuk betiğinde "select" tarafından kullanılan satır

PS3 çevre değişkeni kullanarak aşağıda açıklandığı gibi kabuk betiğinin içerisinde döngü seçmek için özel komut istemcisi tanımlayabilirsiniz. Betikte herhangi bir PS3 değeri belirlemezseniz, select komutu varsayılan olarak komut istemcisinde #? yazacaktır. PS3 değişkeninin değeri select komutu için komut istemcisi olarak kullanılır.

Kabuk betiği ve çıktısı (PS3 olmadan)

```
ramesh@dev-db ~> cat ps3.sh
select i in pzt sal çrş exit
do
  case $i in
    pzt) echo "Pazartesi";;
    sal) echo "Salı";;
    çrş) echo "Çarşamba";;
    exit) exit;;
  esac
done
```

```
ramesh@dev-db ~> ./ps3.sh
1) pzt
2) sal
3) çrş
4) exit
#? 1
Pazartesi
#? 4
```

[Not: Seçilen kabuk istemcisi için varsayılan olarak "#?" gösterir.]

Kabuk betiği ve çıktısı (PS3 ile)

```
ramesh@dev-db ~> cat ps3.sh
PS3="Bir gün seç (1-4): "
select i in pzt sal çrş exit
do
  case $i in
    pzt) echo "Pazartesi";;
    sal) echo "Salı";;
    çrş) echo "Çarşamba";;
    exit) exit;;
  esac
done
```

```
ramesh@dev-db ~> ./ps3.sh
1) pzt
2) sal
3) çrş
4) exit
Bir gün seç (1-4): 1
Pazartesi
Bir gün seç (1-4): 4
```

[Not: Komut istemcisi değiştirilmiş olarak "Bir gün seç (1-4):" şeklinde gösterir.]

İpucu 32. PS4 – set -x kullanıldığında çıktının ön ekini düzenleme

Aşağıdaki gibi kabuk betiğini hata ayıklama modunda çalıştırdığınızda, komut istemcisinde tanımlı PS4 kabuk değişkeni gösterilir.

Kabuk Betiği ve Çıktı (PS4 Olmadan)

```
ramesh@dev-db ~> cat ps4.sh
set -x
echo "PS4 demo betiği"
ls -l /etc/ | wc -l
du -sh ~
ramesh@dev-db ~> ./ps4.sh
++ echo 'PS4 demo betiği'
PS4 demo betiği
++ ls -l /etc/
++ wc -l
243
++ du -sh /home/ramesh
48K /home/ramesh
```

[Not: set -x kullanarak varsayılan olarak komut istemcisinde ++ kullanır.]

Kabuk Betiği ve Çıktı (PS4 ile)

Sıradaki iki kod ps4.sh içinde PS4'ü aşağıdaki gibi tanımlar.

o \$0 - betiğin adını belirtir

o \$LINENO - betiğin içindeki geçerli satır numarasını gösterir

```
ramesh@dev-db ~> cat ps4.sh
PS4='$0.$LINENO+ '
set -x
echo "PS4 demo betiği"
ls -l /etc/ | wc -l
du -sh ~
ramesh@dev-db ~> ./ps4.sh
./ps4.sh.3+ echo 'PS4 demo betiği'
PS4 demo betiği
./ps4.sh.4+ ls -l /etc/
./ps4.sh.4+ wc -l
243
./ps4.sh.5+ du -sh /home/ramesh
48K /home/ramesh
```

[Not: Değiştirilmiş olarak istemcide "{betik-adı}.{satır-no}+ " kullanır.]

İpucu 33. PROMPT_COMMAND

Bash çalıştırıldığında PROMPT_COMMAND 'ın içeriği önceki PS1 değişkenini gösterir. PROMPT_COMMAND, PS1 değişkeninde çalıştırılması istenen komuttur.

```
ramesh@dev-db ~> export PROMPT_COMMAND="date +%k:%m:%S"
22:08:42
ramesh@dev-db ~>
```

[Not: PROMPT_COMMAND ve PS1 değişkenini farklı satırlarda gösterir.]

Eğer PS1 ile PROMPT_COMMAND değerini aynı satırda göstermek isterseniz, aşağıdaki gibi echo -n kullanabilirsiniz.

```
ramesh@dev-db ~> export PROMPT_COMMAND="echo -n [$(date +%k:%m:%S)]"
[22:08:51]ramesh@dev-db ~>
[Not: PROMPT_COMMAND ve PS1 çıktısını aynı satırda gösterir.]
```

Bölüm 6: PS1 kullanarak renkli ve fonksiyonel komut satırı

İpucu 34. Komut istemcisinde kullanıcı , bilgisayar ve izin adı

Bu örnekte PS1, komut satırında aşağıdaki üç bilgiyi gösterir.

- o \u - Kullanıcı adı (Username)
- o \h - Bilgisayar adı (Hostname)
- o \W - Geçerli dizinin adı

```
-bash-3.2$ export PS1="\u@\h \W> "
ramesh@dev-db ~> cd /etc/mail
ramesh@dev-db mail>
```

İpucu 35. Komut istemcisinde şu anki saati gösterme

PS1 çevre değişkeninde, \$(linux_komutu) formatında belirterek herhangi bir linux komutunu direkt olarak çalıştırabilirsiniz. Aşağıdaki örnekte \$(date) komutu, komut istemcisinde şu anki zamanı göstermek için çalıştırıldı.

```
ramesh@dev-db ~> export PS1="\u@\h [$(date +%k:%m:%S)]> "
ramesh@dev-db [11:09:56]>
```


Ayrıca \t kullanarak, şu anki saati ss:dd:ss formatında aşağıdaki gibi gösterebilirsiniz.

```
ramesh@dev-db ~> export PS1="\u@\h [\t]> "
ramesh@dev-db [12:42:55]>
```

Ya da \@ kullanarak, şu anki saati 12 saat öö/ös formatında aşağıdaki gibi gösterebilirsiniz.

```
ramesh@dev-db ~> export PS1="[\@] \u@\h> "
[04:12 PM] ramesh@dev-db>
```

İpucu 36. Komut istemcisinde herhangi bir komutun çıktısını gösterme

Herhangi bir Linux komutunun çıktısını komut istemcisinde gösterebilirsiniz.

Aşağıdaki örnekte komut istemcisinde | (pipe) ile ayrılmış 3 madde gösterilir.

- o \!: Komutun geçmiş komutlardaki sırası
- o \h: Bilgisayar adı (hostname)
- o \$kernel_version: uname -r komutunun çıktısı atanılan değişken
- o \\$?: Son komutun durumu

```
ramesh@dev-db ~> kernel_version=$(uname -r)
ramesh@dev-db ~> export PS1="\!|\h|$kernel_version|\$?> "
473|dev-db|2.6.25-14.fc9.i686|0>
```

İpucu 37. Komut istemcisinde yazıların rengini değiştirme

Komut istemcisini mavi renkte görüntüleme, kullanıcı adı bilgisayar adı ve geçerli dizin bilgisi...

```
$ export PS1="\e[0;34m\u@\h \w> \e[m "
[Not: Bu, komut istemcisinin açık mavi olması için.]
$ export PS1="\e[1;34m\u@\h \w> \e[m "
[Not: Bu, komut istemcisinin koyu mavi olması için.]
```

- o \e[- Renkli komut istemcisinin başlangıcını belirtir.
- o x;ym - Renk kodunu belirtir. İlgili renk kodu değerini aşağıda bulabilirsiniz.
- o \e[m - Renkli komut istemcisinin bitişini belirtir.

Renk Kodu Tablosu:

```
Siyah 0;30
Mavi 0;34
Yeşil 0;32
Camgöbeği (Cyan) 0;36
Kırmızı 0;31
Mor 0;35
Kahverengi 0;33
[Not: 0 ile 1 arasında rengin koyuluğu ayarlanabilir.]
```

Renk değişimini kalıcı yapmak için ~/.bash_profile veya ~/.bashrc içine aşağıdaki satırları ekleyin.

```
$ vi ~/.bash_profile
STARTCOLOR='\e[0;34m';
ENDCOLOR="\e[0m"
export PS1="$STARTCOLOR\u@\h \w> $ENDCOLOR"
```

İpucu 38. Komut istemcisinde arkaplan rengini değiştirme

Aşağıdaki gibi PS1 komut istemcisinde \e[{kod}m belirterek arkaplan rengini değiştirebilirsiniz.

```
$ export PS1="\e[47m\u@\h \w> \e[m "
[Not: Bu, açık gri arkaplan rengi için.]
```

Arkaplan ve önplan rengini değiştirmeyi birleştirme

```
$ export PS1="\e[0;34m\e[47m\u@\h \w> \e[m "
[Not: Açık mavi önplan ve Açık gri arkaplan rengi için.]
```

Aşağıdaki satırları, ~/.bash_profile veya ~/.bashrc dosyanıza ekleyerek üstteki ayarları kalıcı hâle getirebilirsiniz.

```
$ vi ~/.bash_profile
STARTFGCOLOR='\e[0;34m';
STARTBGCOLOR="\e[47m"
ENDCOLOR="\e[0m"
export PS1="$STARTFGCOLOR$STARTBGCOLOR\u@\h \w>
$ENDCOLOR"
```

Aşağıdaki arkaplan renklerinden zevkinize uyacak bir taneyi seçin ve uygulayın.

- o \e[40m
- o \e[41m
- o \e[42m
- o \e[43m
- o \e[44m
- o \e[45m
- o \e[46m
- o \e[47m

İpucu 39. Komut istemcisinde çoklu renk gösterme

Aynı komut istemcisinde çoklu renk gösterebilirsiniz. Aşağıdaki fonksiyonu ~/.bash_profile dosyanıza ekleyin.

```
function prompt {
    local BLUE="\[\033[0;34m\"
    local DARK_BLUE="\[\033[1;34m\"
    local RED="\[\033[0;31m\"
    local DARK_RED="\[\033[1;31m\"
    local NO_COLOR="\[\033[0m\"
    case $TERM in
        xterm*|rxvt*)
            TITLEBAR='\[\033]0;\u@\h:\w\007\'
            ;;
        *)
            TITLEBAR=""
            ;;
    esac
    PS1="\u@\h [\t]> "
    PS1="${TITLEBAR}\
$BLUE\u@\h $RED[\t]>$NO_COLOR "
    PS2='continue-> '
    PS4='$0.$LINENO+ '
}
```

Değişikliklerin etkili olması için tekrar giriş yapmanız ya da .bash_profile dosyasını aşağıda gösterildiği gibi yazmanız gerekir.

```
$. ~/.bash_profile
$ prompt
ramesh@dev-db [13:02:13]>
```

İpucu 40. tput kullanarak komut istemcisinde renk deęiřtirme

Ayrıca ařaęıda gsterildięi gibi tput kullanarak PS1 komut istemcisinin rengini deęiřtirebilirsiniz.

```
$ export PS1="\[$(tput bold)$(tput setb 4)$(tput setaf 7)\]  
\u@\h:\w $ \[$(tput sgr0)\]"
```

tput Renk Durumları:

- o tput setab [1-7] - Kaçış karakterleri kullanarak arkaplan rengini ayarlar
- o tput setb [1-7] - Arkaplan rengini ayarlar
- o tput setaf [1-7] - Kaçış karakterleri kullanarak önplan rengini ayarlar
- o tput setf [1-7] - Önplan rengini ayarlar

tput Text Modu Durumları:

- o tput bold - Kalın moda ayarlar
- o tput dim - Yarı parlak mod açar
- o tput smul - Altıçizili mod başlatır
- o tput rmul - Altıçizili moddan çıkar
- o tput rev - Tersine moda çevirir
- o tput smso - Standout moda girer
- o tput rmso - Standout moddan çıkar
- o tput sgr0 - Tüm özellikleri kapatır

tput için Renk Kodları:

- o 0 - Siyah
- o 1 - Kırmızı
- o 2 - Yeşil
- o 3 - Sarı
- o 4 - Mavi
- o 5 - Eflatun (Magenta)
- o 6 - Camgöbeęi (Cyan)
- o 7 - Beyaz

İpucu 41. PS1 değişkeni için mevcut kodları kullanarak kendi komut istemcisini yaratma

Aşağıdaki kodları kullanarak fonksiyonel ve zevkinize uygun kendi kişisel PS1 Linux komut istemcinizi yaratabilirsiniz.

- o \a => Bir ASCII bell karakteri (07)
- o \d => "Gün Ay Tarih" formatında tarih (örneğin "Sal May 26")
- o \D{format} => strftime(3) çıktısını komut istemcisinde gösterecektir; format boş bırakılırsa yerel saat gösterilir. Parantez gerekmektedir.
- o \e => Bir ASCII kaçış karakteri (033)
- o \h => Bilgisayar adının ilk bölümü
- o \H => Bilgisayar adı
- o \j => Kabuk tarafından şu anda yönetilen toplam görev sayısı
- o \l => Kabuğun terminalinde makine adının dizini
- o \n => Yeni satır
- o \r => Satırbaşı
- o \s => Kabuğun adı, \$0 'ın dizin adı (son eğik çizgiden sonraki kısım)
- o \t => 24-saat ve SS:DD:SS formatında şu anki zaman
- o \T => 12-saat SS:DD:SS formatında şu anki zaman
- o \@ => 12-saat öö/ös (am/pm) formatında şu anki zaman
- o \A => 24-saat SS:DD formatında şu anki zaman
- o \u => Geçerli kullanıcının kullanıcı adı
- o \v => Bash'ın versiyonu (örneğin, 2.00)
- o \V => Bash'ın sürümü, versiyon + yama seviyesi (örn. 2.00.0)
- o \w => Geçerli dizin, \$HOME değişkeninin kısaltılmışı tilde ile
- o \W => Geçerli dizinin anadizin adı, \$HOME değişkeninin kısaltılmışı tilde ile gösterilir
- o \! => Komutun geçmişteki sırası
- o \# => Komutun komut numarası
- o \\$ => Etkin UID(kullanıcı id'si) 0 ise #, aksi takdirde \$
- o \nnn => Sekizli nnn sayısına karşılık gelen karakter
- o \& => Ters eğik çizgi

- o \[=> Yazdırılmayacak karakterler dizisinin başlangıcı
- o \] => Yazdırılmayacak karakterler dizisinin bitişi

İpucu 42. PS1 değişkeni içerisinde bash kabuk fonksiyonu kullanma

Ayrıca, aşağıdaki gibi PS1'de kabuk betik fonksiyonu çalıştırabilirsiniz.

```
ramesh@dev-db ~> function httpdcount {
> ps aux | grep httpd | grep -v grep | wc -l
> }
ramesh@dev-db ~> export PS1="\u@\h [\`httpdcount`]> "
ramesh@dev-db [12]>
```

[Not: Bu, çalışan toplam httpd işlemlerinin sayısını gösterir.]

Aşağıdaki satırı sizin ~/.bash_profile veya ~/.bashrc dosyanıza eklerseniz bu ayarı kalıcı yaparsınız.

```
$ vi .bash_profile
function httpdcount {
ps aux | grep httpd | grep -v grep | wc -l
}
export PS1='\u@\h [\`httpdcount`]> '
```

İpucu 43. PS1 değişkeni içerisinde kabuk betiği kullanmak

Ayrıca, PS1 değişkeninde kabuk betiği de çağırabilirsiniz. Aşağıdaki örnekte, PS1 değişkeni içerisinde çağırılan ~/bin/totalfilesize.sh adlı betik geçerli dizindeki toplam dosya boyutunu hesaplar.

```
ramesh@dev-db ~> cat ~/bin/totalfilesize.sh
for filesize in $(ls -l . | grep "^-" | awk '{print $5}')
do
    let totalsize=$totalsize+$filesize
done
echo -n "$totalsize"

ramesh@dev-db ~> export PATH=$PATH:~/bin
ramesh@dev-db ~> export PS1="\u@\h [\$(totalfilesize.sh) bytes]> "
ramesh@dev-db [534 bytes]> cd /etc/mail
ramesh@dev-db [167997 bytes]>
```

[Not: Bu, totalfilesize.sh betiğini çalıştırıp toplam dosya boyutunu PS1 istemcisinde gösterir.]

Bölüm 7: Arşiv ve Sıkıştırma

İpucu 44. Temel zip komutları

Çoklu dosyalar nasıl sıkıştırılır ?

```
Söz dizimi: zip {.zip uzantılı dosya adı} {sıkıştırılacak dosyalar}
```

```
# zip var-log-files.zip /var/log/*
adding: var/log/acpid (deflated 81%)
adding: var/log/anaconda.log (deflated 79%)
adding: var/log/anaconda.syslog (deflated 73%)
adding: var/log/anaconda.xlog (deflated 82%)
adding: var/log/audit/ (stored 0%)
adding: var/log/boot.log (stored 0%)
adding: var/log/boot.log.1 (deflated 40%)
adding: var/log/boot.log.2 (deflated 42%)
adding: var/log/boot.log.3 (deflated 40%)
adding: var/log/boot.log.4 (deflated 40%)
```

Bir dosya ve onun dosyaları özyinelemeli bir şekilde nasıl sıkıştırılır ?

```
# zip -r var-log-dir.zip /var/log/
updating: var/log/ (stored 0%)
adding: var/log/wtmp (deflated 78%)
adding: var/log/scrollkeeper.log (deflated 94%)
adding: var/log/rpmpkgs.3 (deflated 68%)
adding: var/log/spooler (stored 0%)
adding: var/log/cron.2 (deflated 90%)
adding: var/log/spooler.1 (stored 0%)
adding: var/log/spooler.4 (stored 0%)
adding: var/log/httpd/ (stored 0%)
adding: var/log/rpmpkgs.1 (deflated 68%)
adding: var/log/anaconda.log (deflated 79%)
adding: var/log/secure.2 (deflated 93%)
```

*.zip ifadesine uyan sıkıştırılmış dosyalar nasıl çıkartılır ?

```
# unzip var-log.zip
Archive: var-log.zip
  inflating: var/log/acpid
  inflating: var/log/anaconda.log
  inflating: var/log/anaconda.syslog
  inflating: var/log/anaconda.xlog
   creating: var/log/audit/
```

Çıkarma esnasını detaylı görüntülemek için aşağıdaki gibi -v seçeneğini kullanabilirsiniz.

```
# unzip -v var-log.zip
Archive: var-log.zip
  Length  Method Size   Ratio    Date    Time    CRC-32
  Name
-----
   1916  Defl:N  369    81%    02-08-08 14:27 e2ffdc0c
var/log/acpid
  13546  Defl:N  2900   79%    02-02-07 14:25 34cc03a1
var/log/anaconda.log
skip..
   7680  Defl:N  411    95%    12-30-08 10:55 fe876ee9
var/log/wtmp.1
   40981  Defl:N  7395   82%    02-08-08 14:28 6386a95e
var/log/Xorg.0.log
-----
41406991          2809229   93%          56
files
```

Zip dosyasını çıkarırken içeriği nasıl listelenir ?

```
# unzip -l var-log.zip
Archive: var-log.zip
  Length    Date    Time    Name
-----
   1916    02-08-08  14:27  var/log/acpid
  13546    02-02-07  14:25  var/log/anaconda.log
..skip..
   40981    02-08-08  14:28  var/log/Xorg.0.log
   40981    02-08-07  14:56  var/log/Xorg.0.log.old
-----
41406991          56 files
```


İpucu 45. Zip komutu kullanarak gelişmiş sıkıştırma

Zip komutu tarafından sağlanan 10 sıkıştırma seviyesi vardır.

- o Seviye 0 , herhangi bir sıkıştırma yapmadan sadece arşivleyen en düşük seviyedir.
 - o Seviye 1, küçük ve hızlı bir sıkıştırma uygulayacaktır.
 - o Seviye 6, varsayılan sıkıştırma seviyesidir.
 - o Seviye 9, en yüksek sıkıştırma seviyesidir. Sıkıştırma, varsayılan seviyeye göre daha yavaştır.
- Bence, büyük bir dosya sıkıştırmadıkça, her zaman seviye 9'u kullanın.

Aşağıdaki örnekte, aynı dizini seviye 0, 6 ve 9 ile sıkıştırdım. Sıkıştırılmış dosyanın boyutlarına dikkat edebilirsiniz.

```
# zip var-log-files-default.zip /var/log/*
# zip -0 var-log-files-0.zip /var/log/*
# zip -9 var-log-files-9.zip /var/log/*
# ls
-rw-r--r-- 1 root root 2817248 Jan 1 13:05 var-log-files-default.zip
-rw-r--r-- 1 root root 41415301 Jan 1 13:05 var-log-files-0.zip
-rw-r--r-- 1 root root 2582610 Jan 1 13:06 var-log-files-9.zip
```

İpucu 46. Şifre korumalı zip dosyaları

Zip komutuna verilen -P parametresi ile zip dosyasına bir şifre atayabilirsiniz.

```
# zip -P benimparolam var-log-protected.zip /var/log/*
```

Eğer arkaplandaki işlemler için kabuk betiğinin içinde komut kullanacaksanız, yukarıdaki seçenek iyidir. Ancak, komut satırında etkileşimli sıkıştırma yaparken girdiğiniz şifrenin geçmişte görünmesini istemiyor olabilirsiniz. Bu yüzden, aşağıda gösterildiği gibi -e parametresini kullanarak şifre atayabilirsiniz.

```
# zip -e var-log-protected.zip /var/log/*
Parola girin:
Parolayı doğrulayın:
updating: var/log/acpid (deflated 81%)
updating: var/log/anaconda.log (deflated 79%)
```

Şifre korumalı bir dosyayı çıkarırken, aşağıdaki gibi şifreyi soracaktır.

```
# unzip var-log-protected.zip
Archive: var-log-protected.zip
[var-log-protected.zip] var/log/acpid parola:
```

İpucu 47. Zip arşivi kontrol etmek

Bazen, bir zip arşivini çıkarmadan önce kontrol etmek isteyebilirsiniz. Zip dosyasını test etmek için, aşağıdaki gibi -t parametresini kullanabilirsiniz.

```
# unzip -t var-log.zip
Archive: var-log.zip
testing: var/log/acpid OK
testing:var/log/anaconda.log OK
testing: var/log/anaconda.syslog OK
.....
testing: var/log/wtmp OK
testing: var/log/wtmp.1 OK
testing: var/log/Xorg.0.log OK
var-log.zip arşivinde hiçbir hata tespit edilmedi.
```

İpucu 48. Temel tar komutları

Tar komutu (Tape ARchive), bir grup dosyayı arşive dönüştürmek için kullanılır.

```
Söz dizimi: tar [seçenekler] [tar-arşiv-adı] [diğer-dosya-adları]
```

Ev dizininin altındaki tüm dosya ve altdizinlere nasıl bir yedek dosyası yaratabilirim ?

Aşağıdaki komut, /tmp altında my_home_directory.tar adında bir yedek arşiv dosyası yaratır. Bu arşiv, /home/jsmith dizini altındaki tüm dosya ve altdizinleri içerir.

o c parametresi, bir arşiv yaratmaya yarar. (create)

o v parametresi ayrıntılı modu(verbose mode) belirtir, komut çalıştırılırken ek bilgileri gösterir.

o f parametresi, arşiv dosyasının adının komutta adı geçen dosya olduğunu belirtir.

```
# tar cvf /tmp/my_home_directory.tar /home/jsmith
```

Tar arşivinin içindeki tüm dosyaları nasıl görüntülerim ?

t seçeneği, tar arşivindeki tüm dosyaları gösterecektir.

```
# tar tvf /tmp/my_home_directory.tar
```

Bir tar arşivinden tüm dosyaları nasıl çıkartabilirim ?

Aşağıda gösterildiği gibi x parametresi, tar arşivindeki tüm dosyaları çıkaracaktır. Komut hangi dizinde çalıştırıldıysa, tar arşivinin içeriği de o dizine çıkarılacaktır.

```
# tar xvf /tmp/my_home_directory.tar
```

tar.gz dosyalarını nasıl özel bir dizine çıkartabilirim ?

```
# tar xvfz /tmp/my_home_directory.tar.gz -C /home/ramesh
```

İpucu 49. gzip ve bzip2'yi tar ile birleştirme

gzip, tar ile nasıl kullanılır ?

Sıkıştırılmış dosya tar.gz ile ilgili olduğunda, tar komutuna z seçeneğini ekleyin.

```
# tar cvfz /tmp/my_home_directory.tar.gz /home/jsmith  
# tar xvfz /tmp/my_home_directory.tar.gz  
# tar tvfz /tmp/my_home_directory.tar.gz
```

Not: Gzip kullanmak, bzip2'ye kıyasla daha hızlıdır.

bzip2, tar ile nasıl kullanılır ?

Sıkıştırılmış dosya tar.bz2 ile ilgili olduğunda, tar komutuna j seçeneğini ekleyin.

```
# tar cvfj /tmp/my_home_directory.tar.bz2 /home/jsmith  
# tar xvfj /tmp/my_home_directory.tar.bz2  
# tar tvfj /tmp/my_home_directory.tar.bz2
```

Not: bzip2 kullanmak, gzip kullanmaya kıyasla daha yüksek seviyede sıkıştırma sağlar.

Bölüm 8: Komut Satırı Geçmişi

Linux komut satırını sıklıkla kullanıyorsanız, geçmişte etkili bir şekilde kullanmak büyük bir verim artışı sağlar. Aslında, burda verilen 15 örnekte bir kez ustalaşırsanız, komut satırı kullanmayı daha zevkli ve eğlenceli bulacaksınız.

İpucu 50. HISTTIMEFORMAT ile geçmişte komutun zamanını göstermek

Komut satırına history yazdığınızda tipik olarak komut ve komut sırasını alırsınız. Denetim amacıyla, aşağıda gösterildiği gibi komutun zamanını göstermek yararlı olabilir.

```
# export HISTTIMEFORMAT='%F %T '
# history | more
 1 2008-08-05 19:02:39 service network restart
 2 2008-08-05 19:02:39 exit
 3 2008-08-05 19:02:39 id
 4 2008-08-05 19:02:39 cat /etc/redhat-release
```

Not: Ayrıca aşağıdaki gibi geçmiş x adet komutu görüntülemek için takma ad (alias) tanımlayabilirsiniz.

```
alias h1='history 10'
alias h2='history 20'
alias h3='history 30'
```

İpucu 51. Ctrl+R kullanarak geçmişte arama yapma

Kuvvetle inanıyorum ki, geçmişte en fazla kullandığınız özellik bu olacaktır. Önceden çalıştırdığınız uzun bir komut olduğunda, kelimeleri girip geçmişte basitçe bir araştırma yapabilir ve aynı komutu hepsini yazmadan tekrar çalıştırabilirsiniz. Ctrl+R'ye basın ve sözcükleri yazın.

Aşağıdaki örnekte, red kelimesini aradığımda geçmişte içeriğinde red kelimesi geçen “cat /etc/redhat-release” komutunu gösterdi.

```
# [Not: Komut istemcisinde Ctrl+R'ye bastığında, aşağıdaki gibi istemcide reverse-i-search yazacaktır.]

(reverse-i-search)`red': cat /etc/redhat-release
[Not: Gördüğünüz komutta enter'a bastığınızda komut geçmişten çalıştırılacaktır.]

# cat /etc/redhat-release
Fedora release 9 (Sulphur)
```

Bazen, komutu geçmişten çalıştırmadan önce düzenlemek isteyebilirsiniz. Örneğin, httpd aradığınızda, geçmişte service httpd stop çıkabilir, aşağıdaki gibi stop'u start yapıp komutu tekrar çalıştırabilirsiniz.

```
# [Not: Komut istemcisinde Ctrl+R'ye bastığında, aşağıdaki gibi istemcide reverse-i-search yazacaktır.]

(reverse-i-search)`httpd': service httpd stop
[Not: Sağ ya da sol ok'a bastığınızda komutu istemcide göreceksiniz. Çalıştırmadan önce istediğiniz gibi düzenleyebilirsiniz.]

# service httpd start
```

İpucu 52. Dört farklı yöntem kullanarak hızlı bir şekilde önceki komutu tekrar etme

Bazen, çeşitli nedenlerle önceki komutları tekrar etmek isteyebilirsiniz. Aşağıdaki 4 farklı yol, son çalıştırılan komutu tekrar etmeye yarar.

1. Yukarı tuşunu kullanarak önceki komutu görüntüleyebilir ve enter'a basarak tekrar çalıştırabilirsiniz.
2. !! yazıp enter'a basabilirsiniz.
3. !-1 yazıp enter'a basabilirsiniz.
4. Ctrl+P'ye basarsanız önceki komut görüntülenir ve enter'a basıp çalıştırabilirsiniz.

İpucu 53. Geçmişten özel komut çalıştırma

Aşağıdaki örnekte, eğer 4 numaralı komutunu tekrar etmek istiyorsanız, aşağıdaki gibi !4 çalıştırmalısınız.

```
# history | more
  1 service network restart
  2 exit
  3 id
  4 cat /etc/redhat-release

# !4
cat /etc/redhat-release
Fedora release 9 (Sulphur)
```

İpucu 54. Özel kelime ile başlayan önceki komutu çalıştırma

! yazıp çalıştırmak istediğiniz komutun baştan birkaç karakterini yazarsanız komutu tekrarlayabilirsiniz. Aşağıdaki örnekte, !ps yazıp enter'a basılarak önceki komutlardan(sondan başa doğru) ps ile başlayan 'ps aux | grep yp' komutu çalıştırılmıştır.

```
# !ps
ps aux | grep yp
  root   16947   0.0   0.1  36516   1264  ?
Sl   13:10   0:00  ypbind
  root   17503   0.0   0.0   4124    740  pts/0
S+   19:19   0:00  grep yp
```

İpucu 55. HISTSIZE kullanarak geçmişteki toplam satır sayısını kontrol etme

Aşağıdaki satırları .bash_profile dosyanıza ekleyin ve bash kabuğuna tekrar giriş yapın. Örnekte, geçmişte sadece 450 komut saklanacaktır.

```
# vi ~/.bash_profile

HISTSIZE=450
HISTFILESIZE=450
```

İpucu 56. HISTFILE kullanarak geçmiş dosyası adı değiştirme

Varsayılan olarak, geçmiş ~/.bash_history dosyasında saklanır. Geçmişin, .bash_history dosyası yerine .commandline_warrior adlı dosyada depolanması için aşağıdaki satırı .bash_profile dosyanıza ekleyin ve bash kabuğuna tekrar giriş yapın. Bunun pratik bir kullanım olup olmadığını anlamış değilim. Farklı terminaller farklı geçmiş dosya adları kullanabilir.

```
# vi ~/.bash_profile
HISTFILE=/root/.commandline_warrior
```

İpucu 57. HISTCONTROL kullanarak geçmişte sürekli tekrarlanan girdileri elemek

Aşağıdaki örnekte pwd üç kere yazıldı, history yazdığınızda, üç tane aynı komuttan göreceksiniz. Çiftleri elemek için, HISTCONTROL'u çiftleri görmeyecek şekilde ignoredups yazarak ayarlayabilirsiniz.

```
# pwd
# pwd
# pwd
# history | tail -4
 44 pwd
 45 pwd
 46 pwd
 47 history | tail -4
```

[Not: Yukarıdaki gibi pwd komutunu üç kere çalıştırdıktan sonra, geçmişte üç adet pwd komutu var.]

```
# export HISTCONTROL=ignoredups
# pwd
# pwd
# pwd
# history | tail -3
 56 export HISTCONTROL=ignoredups
 57 pwd
 58 history | tail -4
```

[Not: Yukarıdaki gibi pwd komutu üç kere çalıştırıldıktan sonra, geçmişte yalnız bir tane pwd komutu var.]

İpucu 58. HISTCONTROL kullanarak geçmişteki tekrarlanmış komutları silmek

Yukarıda gösterildiği gibi ignoredups sadece ardışık komutları siler. Tüm geçmiş genelinde tekrarlanan komutları silmek için, aşağıdaki gibi HISTCONTROL'u erasedups şeklinde ayarlayabilirsiniz.

```
# export HISTCONTROL=erasedups
# pwd
# service httpd stop
# history | tail -3
    38 pwd
    39 service httpd stop
    40 history | tail -3
# ls -ltr
# service httpd stop
# history | tail -6
    35 export HISTCONTROL=erasedups
    36 pwd
    37 history | tail -3
    38 ls -ltr
    39 service httpd stop
    40 history | tail -6
```

[Not: Önceki 'service httpd stop' silindi.]

İpucu 59. HISTCONTROL ile özel bir komut kullanarak zorunlu olarak geçmişi hatırlamamak

Bir komut çalıştırdığınızda, aşağıdaki gibi komutun önünde boşluk kullanıldığında HISTCONTROL ayarını kullanarak boşluk olan komutu görmezden gelebilirsiniz. Geçmişten komut gizleme olayı, bir çok genç sistem yöneticisini heyecanlandırabilir.

Bu, boşluk gizlemenin(ignoreospace) nasıl çalıştığı için iyi bir örnektir.

```
# export HISTCONTROL=ignoreospace
# ls -ltr
# pwd
# service httpd stop
```

[Not: Service'in başında boşluk olduğu için bu komut geçmişe kaydedilmeyecek.]


```
# history | tail -3
67 ls -ltr
68 pwd
69 history | tail -3
```

İpucu 60. -c parametresi ile tüm geçmişi temizlemek

Bazen, tüm geçmişi temizlemek isteyebilirsiniz. Ancak, sonraki komutların tutulmaya devam etmesi gerekir.

```
# history -c
```

İpucu 61. Geçmiş komutun parametresini yeni bir komuta verme

Geçmişte aradığınızda bulduğunuz komutun parametresini farklı bir komutta kullanmak isteyebilirsiniz. Aşağıdaki örnekte, !:\$ ibaresi önceki komutun parametresini sonraki vi komutuna verir.

```
# ls anaconda-ks.cfg
anaconda-ks.cfg
# vi !:$
vi anaconda-ks.cfg
```

Aşağıdaki örnekte, !^ ibaresi önceki komutun ilk parametresini sonraki komuta verir.

```
# cp anaconda-ks.cfg anaconda-ks.cfg.bak
anaconda-ks.cfg
# vi !^
vi anaconda-ks.cfg
```

İpucu 62. Geçmiş komutun istenilen parametresini yeni komuta vermek

Aşağıdaki örnekte, !cp:2 geçmişte cp ile başlayan komutları araştırır ve ilgili komutun 2. parametresini alıp ls -l komutunun parametresi olarak verir.

```
# cp ~/longname.txt /really/a/very/long/path/long-filename.txt
# ls -l !cp:2
ls -l /really/a/very/long/path/long-filename.txt
```

Aşağıdaki örnekte, !cp:\$ geçmişte cp ile başlayan komutları araştırır ve ilgili komutun son argümanını(\$ bunu belirtir) ls -l'ye argüman olarak verir.

```
# ls -l !cp:$  
ls -l /really/a/very/long/path/long-filename.txt
```

İpucu 63. HISTSIZE kullanarak geçmişi devredışı bırakmak

Eğer geçmişi tamamen devredışı bırakmak istiyorsanız ve yazdığınız komutları hatırlamayacağınız kabuk betiğiyle uğraşmak istemiyorsanız, aşağıdaki gibi HISTSIZE değerini 0 yapabilirsiniz.

```
# export HISTSIZE=0  
# history  
# [Not: history komutu çıktı vermeyecektir.]
```

İpucu 64. HISTIGNORE kullanarak özel komutların geçmişte kayıtlanmasını sağlamak

Bazen, geçmişin pwd ve ls gibi temel komutlarla dağınık görünmesini istemeyebilirsiniz. HISTIGNORE kullanarak, geçmişe kaydedilmesini istemediğiniz tüm komutları belirtebilirsiniz.

HISTIGNORE'a sadece ls eklemek ls komutunu geçmişe kaydetmez, ls -l komutu geçmişe eklenecektir. Bu yüzden, komut adını tam girmelisiniz.

```
# export HISTIGNORE="pwd:ls:ls -ltr:"  
# pwd  
# ls  
# ls -ltr  
# service httpd stop  
# history | tail -3  
79 export HISTIGNORE="pwd:ls:ls -ltr:"  
80 service httpd stop  
81 history  
[Not: pwd,ls ve ls -ltr komutu geçmişte görüntülenmedi.]
```

Bölüm 9: Sistem Yöneticisi Görevleri

İpucu 65. fdisk ile diski bölümlenme

Sunucunuzda yeni diskler yükledikten sonra, bölümlenmek için fdisk gibi araçları kullanabilirsiniz.

Aşağıdaki 5 tipik olay(komut)'ı fdisk'in içinde çalıştırabilirsiniz.

- o n - Yeni bölüm yaratma
- o d - Bulunan bölümü silme
- o p - Bölümlenme tablosunu yazdırma
- o w - Bölümlenme tablosundaki değişiklikleri yazdırma
- o q - fdisk yardımcı uygulamasından çıkış

Bir bölüm yaratma

Aşağıdaki örnekte, /dev/sda1 birincil bölümünü yarattım.

```
# fdisk /dev/sda

Cihaz,ne geçerli bir DOS bölümlenme tablosu ne de Sun,SGI veya OSF disk
etiketine sahip. Yeni bir DOS disk etiketi yapılandırın. Değişiklikler sadece
siz onu yazmaya karar verene kadar hafızada kalacak. Daha sonra, tabii ki,
önceki içerik geri alınabilir olmayacak.
Bu disk için silindir sayısı 34893 olarak ayarlandı. Bunda yanlış hiçbir şey
yok, ama bu 1024'e göre daha geniş ve belirli kurulumları şu sorunlara neden
olabilir:
1) Önyüklem sırasında çalıştırılan yazılım (ör. Eski LILO sürümleri)
2) Diğer işletim sistemlerinden önyüklem ve bölümlenme yazılımı (ör. DOS FDISK)
Uyarı: Bölümlenme tablo 4'ün 0x0000 geçersiz etiketi w (write - yazılabilir) ta-
rafından düzeltilecektir.

Komut (yardım için m): p
Disk /dev/sda: 287.0 GB, 287005343744 byte
255 kafa, 63 bölüm(sector/track), 34893 silindir
Unit = silindir 16065 * 512 = 8225280 byte

Komut (yardım için m): n
Komut eylemi:
  e genişletilmiş (extended)
  p öncelikli bölümlenme (1-4)
```

p

Bölümlenme numarası (1-4): **1**

İlk silindir (1-34893, varsayılan 1):

Varsayılan değer (1) kullanılıyor

Son silindir veya +size veya +sizeM veya +sizeK (1-34893, varsayılan 34893)

Varsayılan değer (34893) kullanılıyor

Komut (yardım için m): **w**

Bölümlenme tablosu değiştirildi.

Bölümlenme tablosunu tekrar okumak için ioctl() çağrılıyor.

Diskler senkronize ediliyor.

Bölümün başarıyla oluşturulduğunu doğrulama

fdisk /dev/sda

Bu disk için silindir sayısı 34893 olarak ayarlandı. Bunda yanlış hiçbir şey yok, ama bu 1024'e göre daha geniş ve belirli kurulumları şu sorunlara neden olabilir:

1) Önyüklemeye sırasında çalıştırılan yazılım (ör. Eski LILO sürümleri)

2) Diğer işletim sistemlerinden önyüklemeye ve bölümlenme yazılımı (ör. DOS FDISK)

Uyarı: Bölümlenme tablo 4'ün 0x0000 geçersiz etiketi w (write - yazılabilir) tarafından düzeltililecektir.

Komut (yardım için m): **p**

Disk /dev/sda: 287.0 GB, 287005343744 byte

255 kafa, 63 bölüm(sectors/track), 34893 silindir

Unit = silindir 16065 * 512 = 8225280 byte

Başlatılan cihaz	Başla	Bitiş	Blok	Id	Sistem
/dev/sda1	1	34893	280277991	83	Linux

Komut (yardım için m): **q**

İpucu 66. mke2fsk ile diski biçimlendirme

Diskleri bölümlendikten sonra, henüz kullanıma hazır olmayacaklardır. Kullanıma hazır hâle getirmek için biçimlendirmeniz gerekir. Bu aşamada, disk bilgilerini görüntülemeye çalışırsanız, geçerli bir superblok olmadığını belirten aşağıdaki hata iletisini verecektir.

```
# tune2fs -l /dev/sda1
tune2fs 1.35 (28-Şub-2004)
tune2fs: /dev/sda1'i açmaya çalışırken superblokta "bad magic number" hatası
ile karşılaşıldı.
Geçerli dosya sistemi superbloğu bulunamadı.
```

Diski biçimlendirmek için, mke2fs'yi aşağıdaki gibi kullanmalısınız.

```
# mke2fs /dev/sda1
```

Ayrıca, mke2fs'ye aşağıdaki parametreleri de verebilirsiniz.

o -m 0 : Ayrılmış blok yüzdesi - Bu, kök kullanıcı için ayrılmış olan dosya sistemi bloklarının yüzdesini gösterir. Varsayılan %5'tir. Aşağıdaki örnekte 0 olarak ayarlanmıştır.

o -b 4096 : Byte olarak belirtilen blok boyutu. Blok için geçerli değerler 1024,2048 ve 4096 byte'dır.

```
# mke2fs -m 0 -b 4096 /dev/sda1
mke2fs 1.35 (28-Şub-2004)
Dosya sistemi etiketi=
İS tipi: Linux
Blok boyutu=4096 (log=2)
Parça boyutu=4096 (log=2)
205344 inode, 70069497 blok
0 blok (0.00%) süper kullanıcı için ayrıldı
İlk veri bloğu=0
Maximum dosya sistemi bloğu=71303168
2139 blok grupları
32768 grup başına blok, 32768 grup başına parça
96 grup başına inode
Bloklarda depolanan superblok yedekleri:
32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
4096000, 7962624, 11239424, 20480000, 23887872
Inode tablolarını yazdırma: bitti
Superbloklar ve dosya sistemi bilgileri yazdırma: bitti
Bu dosya sistemi her 32 ay veya 180 günde bir denetlenecek. İşlemi geçersiz
kılmak için tune2fs -c veya -i seçeneğini kullanabilirsiniz.
```

Yukarıdaki komut, bir ext2 dosya sistemi yaratacaktır. Bir ext3 dosya sistemi yaratmak için aşağıdakileri yapın:

```
# mkfs.ext3 /dev/sda1  
# mke2fs -j /dev/sda1
```

İpucu 67. Bölümlenmiş diski mount etme

Bir bölüm yaratıp biçimlendirdikten sonra, onu bir bağlama noktasına bağlamak isteyebilirsiniz. Bu bağlama işlemine mount etme denir.

İlk olarak bölümün mount edileceği bir bağlantı noktası(mount point) yani bir klasör oluşturmamız gerekiyor.

```
# mkdir /home/database
```

Dosya sistemini mount etme. (Bir klasöre bağlama)

```
# mount /dev/sda1 /home/database
```

Yeniden başlattıktan sonra otomatik olarak dosya sistemini bağlamak için, aşağıdaki satırı /etc/fstab dosyasına ekleyin.

```
/dev/sdaa /home/database ext3 defaults 0 2
```

İpucu 68. tune2fs ile diske ince ayar yapma

tune2fs -l /dev/sda1 kullanarak dosya sistemi bilgisini aşağıda gösterildiği gibi görüntüleyebilirsiniz.

```
# tune2fs -l /dev/sda1  
tune2fs 1.35 (28-Şub-2004)  
Dosya sistemi birim adı: /home/database  
Son mount etme: <uygun değil>  
Dosya sistemi UUID: f1234556-e123-1234-abcd-bbbbaaaaae11  
Dosya sistemi magic number: 0xEF44  
Dosya sistemi revizyonu#: 1 (dinamik)  
Dosya sistemi özellikleri: resize_inode filetype sparse_super  
Varsayılan mount seçenekleri: (yok)  
Dosya sistemi durumu: temiz değil  
Hatalarda davranış: Devam et  
Dosya sistemi İS tipi: Linux
```

```
Inode sayısı: 1094912
Blok sayısı: 140138994
Ayrılan blok sayısı: 0
Boş blok: 16848481
Boş inode: 1014969
İlk blok: 0
Blok boyutu: 2048
Parça boyutu: 2048
Ayrılmış GDT blokları: 512
Grup başına blok: 16384
Grup başına parça: 16384
Grup başına inode: 128
Grup başına inode bloğu: 8
Dosya sistemi yaratıldı: Sal Tem 1 00:06:03 2008
Son mount zamanı: Per Ağu 21 05:58:25 2008
Son yazma zamanı: Cum Oca 2 15:40:36 2009
Mount sayısı: 2
Maximum mount sayısı: 20
Son kontrol edilme: Sal Tem 1 00:06:03 2008
Kontrol aralığı: 15552000 (6 ay)
Sonraki kontrol: Cmt Ara 27 23:06:03 2008
Ayrılmış blok uid: 0 (kök kullanıcı)
Ayrılmış blok gid: 0 (kök grup)
İlk inode: 11
Inode boyutu: 128
Varsayılan dizin hash: tea
Dizin hash (Directory Hash Seed): 12345829-1236-4123-9aaa-cccc123292b
```

Ayrıca, ext2/ext3 dosya sistemi parametresini ayarlamak için tune2fs'yi de kullanabilirsiniz. Örneğin, dosya sistemi birim adını (filesystem volume name) değiştirmek isterseniz, aşağıda gösterildiği gibi yapabilirsiniz.

```
# tune2fs -l /dev/sda1 | grep volume
Filesystem volume name: /home/database

# tune2fs -L database-home /dev/emcpowera1
tune2fs 1.35 (28-Şub-2004)

# tune2fs -l /dev/sda1 | grep volume
Filesystem volume name: database-home
```

İpucu 69. Takas alanı oluşturma

Takas kullanımını için aşağıda gösterildiği gibi bir dosya oluşturabilirsiniz.

```
# dd if=/dev/zero of=/home/swap-fs bs=1M count=512
512+0 records in
512+0 records out

# ls -l /home/swap-fs
-rw-r--r-- 1 root root 536870912 Oct 2 23:13 /home/swap-fs
```

Yukarıda yaratılan /home/swap-fs dosyasında bir Linux takas alanı kurmak için mkswap kullanabilirsiniz.

```
# mkswap /home/swap-fs
Takas alanı kurma versiyon 1, boyut = 536866 kB
```

Dosya yaratıldıktan ve Linux takas alanı için kurulduktan sonra aşağıda gösterildiği gibi swapon kullanarak takas işlemini etkinleştirebilirsiniz.

```
# swapon /home/swap-fs
```

/etc/fstab dosyasına aşağıdaki satırı ekleyin ve takasın yürürlüğe girmesi için sistemi yeniden başlatın.

```
/home/swap-fs swap swap defaults 0 0
```

İpucu 70. Yeni kullanıcı oluşturma

Yeni kullanıcı ekleme – Temel metod

Yalnızca kullanıcı adı belirtin.

```
# useradd jsmith
```


Yeni kullanıcıyı ek parametreler ile ekleme

Ayrıca, useradd'a aşağıdaki parametreleri de belirtebilirsiniz.

- o -c : Kullanıcı hakkında açıklama.
- o -e : aa/gg/yy formatında kullanıcının son kullanma tarihi

```
# adduser -c "John Smith - Oracle Developer" -e 12/31/09 jsmith
```

Kullanıcının başarıyla eklendiğini doğrulayın.

```
# grep jsmith /etc/passwd
jsmith:x:510:510:John Smith - Oracle Developer:/home/jsmith:/bin/bash
```

Kullanıcı şifresini değiştirme

```
# passwd jsmith
Kullanıcı jsmith için şifre değişiyor
Yeni UNIX şifresi:
KÖTÜ ŞİFRE: Sözlük kelimesine dayanıyor.
Tekrar yeni UNIX şifresi:
passwd:Tüm kimlik doğrulama belirteçleri başarıyla güncellendi.
```

Not: Şifre olarak sadece bir kelime seçtiğimiz için “kötü şifre” hatası aldık. Güçlü şifreler oluşturmak için [link](#)'teki makaleyi okuyun.

Useradd tarafından kullanılan varsayılan değerleri nasıl belirleyebilirim ?

Aşağıda, bir kullanıcı oluşturulduğunda kullanılacak olan varsayılan değerler verilmiştir.

```
# useradd -D
GRUP=100
ANADİZİN=/home
PASİF=-1
BİTİŞ=
KABUK=/bin/bash
SKEL=/etc/skel
```

İpucu 71. Yeni grup oluşturma ve ona bir kullanıcı atama

Yeni bir geliştirici grubu oluşturma

```
# groupadd developers
```

Grubun başarıyla oluşturulduğunu doğrulama

```
# grep developer /etc/group
developers:x:511:
```

Mevcut bir gruba kullanıcı ekleme

Aşağıdaki hatayı alacağınızdan, var olan bir kullanıcıyı değiştirmek için useradd komutunu kullanamazsınız,

```
# useradd -G developers jsmith
useradd: jsmith adında bir kullanıcı var

# usermod -g developers jsmith
```

Kullanıcı gruplarının başarıyla değiştirildiğini doğrulayın

```
# grep jsmith /etc/passwd
jsmith:x:510:511:Oracle Developer:/home/jsmith:/bin/bash

# id jsmith
uid=510(jsmith) gid=511(developers) groups=511(developers)

# grep jsmith /etc/group
jsmith:x:510:
developers:x:511:jsmith
```

İpucu 72. OpenSSH'de SSH parolasız oturum açma ayarları

Uzak bir Linux sunucuya, şifre girmeden ssh-keygen ve ssh-copy-id kullanarak aşağıda açıklandığı gibi üç basit adımla giriş yapabilirsiniz. (ortak anahtar=public key) ssh-keygen, ortak ve özel anahtarlar oluşturur. ssh-copy-id, bilgisayarın genel anahtarını uzak sunucunun authorized_keys (yetkili anahtarlar) dosyasına kopyalar. ssh-copy-id, ayrıca uzak sunucu'nun home klasörüne uygun izinleri atar.

1- Yerel bilgisayarda ssh-keygen kullanarak ortak ve özel anahtar yaratma

```
jsmith@local-host$ ssh-keygen
Genel/Özel RSA anahtar çifti üretiliyor.
Anahtarın kaydedileceği dosyayı girin(/home/jsmith/.ssh/id_rsa):[Enter]
Parolayı girin(boş bırakabilirsiniz): [Enter'a bas]
Tekrar parolayı girin: [Enter'a bas]
Kimliğiniz kaydedildi. /home/jsmith/.ssh/id_rsa.
Herkese açık (ortak) anahtarınız kaydedildi. /home/jsmith/.ssh/id_rsa.pub.
Anahtar parmak izi:
33:b3:fe:af:95:95:18:11:31:d5:de:96:2f:f2:35:f9
jsmith@local-host
```

2- ssh-copy-id kullanarak genel anahtarı uzak sunucuya kopyalama

```
jsmith@local-host$ ssh-copy-id -i ~/.ssh/id_rsa.pub uzak-sunucu
jsmith@uzak-sunucu'nun şifresi:
Şimdi makineye ssh ile giriş yapmayı deneyin ve kaydedin,
Beklemediğiniz ekstra anahtar eklenmediğine emin olmak için
.ssh/authorized_keys dosyasını kontrol edin.
```

Not: ssh-copy-id anahtarları uzak sunucunun .ssh/authorized_keys dosyasına ekler.

3- Uzak sunucuda şifre girmeden oturum açma

```
jsmith@local-host$ ssh uzak-sunucu
Son Giriş: Paz Kas 16 17:22:33 2008 adresinden 192.168.1.2
[Not: SSH şifre istemedi.]
jsmith@uzak-sunucu$ [Not: Burada uzak-sunucu'dasınız.]
```

İpucu 73. ssh-agent ile birlikte ssh-copy-id kullanımı

ssh-copy-id ile birlikte ssh-add/ssh-agent'ı kullanma

-i seçeneği için herhangi bir değer geçmediğinde ve ~/.ssh/identity.pub kullanılamıyorsa, ssh-copy-id aşağıdaki hata mesajını gösterecektir.

```
jsmith@local-host$ ssh-copy-id -i uzak-sunucu
/usr/bin/ssh-copy-id: HATA: Kimlik bulunamadı.
```

Eğer ssh-add kullanarak ssh-agent'a anahtarları yüklediyseniz, ssh-copy-id, uzak sunucuya kopyalayacağı anahtarları ssh-agent'dan alacaktır. Örneğin, ssh-copy-id komutunda -i seçeneği kullanmadığınızda, ssh-add -L komutunun verdiği anahtarları uzak sunucuya kopyalar.

```
jsmith@local-host$ ssh-agent $SHELL
jsmith@local-host$ ssh-add -L
Kimlik bulunamadı.
jsmith@local-host$ ssh-add
Kimlik eklendi: /home/jsmith/.ssh/id_rsa (/home/jsmith/.ssh/id_rsa)
jsmith@local-host$ ssh-add -L
ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAsJIEILxftj8aSxMa3d8t6JvM79D
aHrtPhTYpq7kIEMUNzApyxsHpH1tQ/Ow== /home/jsmith/.ssh/id_rsa
jsmith@local-host$ ssh-copy-id -i uzak-sunucu
jsmith@uzak-sunucu'nun şifresi:
Şimdi makineye, "ssh uzak-sunucu" şeklinde giriş yapmayı deneyin ve .ssh/authorized_keys'e
beklemediğiniz fazladan anahtarları eklediğinizden emin olmak için kontrol edin.
[Not: ssh-add -L ile gösterilen anahtarı ekledi.]
```

ssh-copy-id'nin üç küçük rahatsız edici noktası

Aşağıda, ssh-copy-id'nin birkaç küçük sıkıntısı bulunmaktadır.

1. Varsayılan ortak anahtar: ssh-copy-id, ~/.ssh/identity.pub'ı varsayılan ortak anahtar dosyası olarak kullanır(-i seçeneğine değer vermediğinizde). Bunun yerine id_dsa.pub, id_rsa.pub veya identity.pub kullanmak istiyoruz. Örneğin, bunlardan herhangi biri varsa bunu uzaktaki sunucuya kopyalayacaktır. Eğer ikisi veya üçü varsa, varsayılan olarak identity.pub kopyalayacaktır.
2. Kimlik bulunamadı: ssh-agent çalıştığında ve ssh-add -L "Kimlik bulunamadı" dediğinde (ssh-agent'a anahtar eklenmemişken), ssh-copy-id, yine de "Kimlik bulunamadı" mesajını uzak sunucunun authorized_keys (yetkili anahtarlar) dosyasına kopyalayacaktır.
3. Authorized_keys dosyasına yinelenen giriş: ssh-copy-id, uzak sunucunun yetkili anahtarlarındaki yinelenen girdileri doğrulamak ister. Eğer yerel bilgisayarda (localhost) ssh-copy-id 'yi birden çok kez çalıştırırsanız, kopyaları kontrol etmeden aynı anahtarı uzak sunucunun authorized_keys

dosyasına eklemeye devam edecektir. Çift girişlerde bile her şey beklendiği gibi çalışır. Ama `authorized_keys` dosyası fazlasıyla dağınık olacaktır.

İpucu 74. Crontab

'cron' kullanarak bir kabuk betiğini veya Linux komutunu belirli zaman veya tarihte çalıştırabilirsiniz. Örneğin, bir sistem yöneticisinin her gün yedekleme yapması gerekebilir.

Cron'a bir iş nasıl eklenir ?

```
# crontab -e
0 5 * * * /root/bin/backup.sh
```

Bu, `/root/bin/backup.sh`'yi her gün saat sabah 5'te çalıştıracaktır.

Cron alanlarının tanımı

Aşağıda crontab dosyasının formatı verilmiştir.

{dakika} {saat} {ayın günü} {ay} {haftanın günü} {kabuk betiğinin yolu}

- o Dakika: İzin verilen aralık 0-59
- o Saat: İzin verilen aralık 0-23
- o Ayın günü: İzin verilen aralık 0-31
- o Ay: İzin verilen aralık 1-12. 1 = Ocak. 12 = Aralık.
- o Haftanın günü: İzin verilen aralık 0 - 7. Pazar, hem 0 hem 7'dir.

Crontab örnekleri

1. Her gece yarısından 1 dakika sonra, saat ö.ö 00.01'de çalıştırma. Bu, sistem yük altında değilken yedekleme yapılabilir iyi bir zamandır.

```
1 0 * * * /root/bin/backup.sh
```

2. Her hafta (Pzt-Cum) saat ö.s 11.59'da yedekleme yapma.

```
59 11 * * 1,2,3,4,5 /root/bin/backup.sh
```

Ayrıca aşağıdaki de aynı işlemi yapar.

```
59 11 * * 1-5 /root/bin/backup.sh
```

3. Komutu her 5 dakikada bir çalıştırma.

```
*/5 * * * * /root/bin/check-status.sh
```

4. Her haftanın ilk günü ö.s 1.10'da çalıştırma.

```
10 13 1 * * /root/bin/full-backup.sh
```

5. Haftaiçi ö.s 11'de çalıştırma.

```
0 23 * * 1-5 /root/bin/incremental-backup.sh
```

Crontab seçenekleri

- o crontab -e : Crontab dosyasını düzenler. Eğer bulunmuyorsa yeni bir crontab dosyası yaratır.
- o crontab -l : Crontab dosyasını gösterir.
- o crontab -r : Crontab dosyasını siler.
- o crontab -ir : Bu, bir crontab'ı silmeden önce kullanıcıya soracaktır.

İpucu 75. Magic SysRq Key kullanarak Linuxu güvenli yeniden başlatma

Magic SysRq Key, Linux çekirdeğinde çeşitli alt seviye komutları ne olursa olsun sistemin durumuna uygulayan bir tuş kombinasyonudur.

Genellikle donmalarda veya yeniden başlatmalarda dosya sistemini bozulmadan kurtarmak için kullanılır. Tuş kombinasyonu Alt+SysRq+tuş içerir. Bazı sistemlerde, SysRq tuşu ekran görüntüsü tuşudur. (prt sc)

Öncelikle, aşağıdaki gibi SysRq tuşunu aktif etmeniz lazım.

```
echo "1" > /proc/sys/kernel/sysrq
```

SysRq Komut Tuşlarının Listesi

- o 'k' - Geçerli sanal konsolda çalışan tüm işlemleri bitirir. (kill sinyali)
- o 'b' - Bölümleri kaldırmadan veya senkronize etmeden sistemi derhâl yeniden başlatır.
- o 'e' - init hâricinde tüm işlemlere SIGTERM gönderir.
- o 'm' - Mevcut bellek bilgisini çıktı olarak verir.
- o 'i' - init dışında tüm işlemlere SIGKILL sinyali gönderir.
- o 'r' - Klavyeyi raw mod'dan (X11 gibi programların kullandığı mod), XLATE mod'a değiştirir.
- o 's' - Tüm mount edilmiş dosya sistemini senkronize eder.
- o 't' - Geçerli görevleri ve bilgilerini çıktı olarak verir.
- o 'u' - Tüm mount edilmiş dosya sistemini salt okunur mod'da (readonly mode) tekrar mount eder.
- o 'o' - Sistemi derhâl kapatır.
- o 'p' - Geçerli kayıtları ve bayrakları çıktı olarak verir.
- o '0-9' - Konsol log seviyesini belirler, hangi çekirdek mesajlarının yazdırılacağını kontrol eder.
- o 'f' - Daha fazla bellek gerektiren işlemi sona erdirmek için oom_kill'i çağırır.
- o 'h' - Yardımı görüntülemek için kullanılır. Ama, yukarıda listelenmiş tuşlardan farklı bir tuşa basarak yardım yazdırabilirsiniz.

Bunun için ayrıca /proc/sysrq-trigger dosyasına tuşların yansımalarını yazdırıyoruz. (echo) Örneğin, bir sistemi yeniden başlatmak için aşağıdakini gerçekleştirebilirsiniz.

```
echo "b" > /proc/sysrq-trigger
```

Magic SysRq Key kullanarak Linux'u güvenli yeniden başlatma

Kapatılan bir Linux bilgisayarın güvenli bir şekilde yeniden başlatılmasını sağlamak için aşağıdakileri yapın. Bu, bir dahaki yeniden başlatma sırasında fsck'yı (file system check) önleyecektir. Örneğin, Alt+SysRq+ [aşağıda vurgulanan harf]

- o un[R]aw (X11'den klavyenin kontrolünü geri al),
- o t[E]rminate (Tüm işlemlere SIGTERM gönder, zatifçe sonlandırmalarına izin ver),
- o k[I]ll (Tüm işlemlere SIGKILL gönderir, onları sonlandırmalarına zorla),
- o [S]ync (Verileri diske boşalt),
- o [U]nmount (Tüm dosya sistemini salt okunur olarak tekrar mount et),
- o re[B]oot. (Yeniden başlat)

Bölüm 10: Apachectl ve Httpd Örnekleri

Apache2'yi yükledikten sonra, eğer apachectl ve httpd'yi maksimum potansiyelde kullanmak istiyorsanız start,stop ve restart'ın ötesine geçmelisiniz. Bu bölümde apachectl ve httpd'yi etkili bir biçimde nasıl kullanacağınız konusunda yardımcı olan 9 örnek gösterilmiştir.

Apachectl, SysV init betikleri gibi start,stop,restart ve status argümanları alır.Ayrıca apachectl, httpd'nin ön-plan hâli denilebilir. Bu yüzden, apachectl kullanarak tüm httpd komutlarını çalıştırabiliriz.

İpucu 76. Farklı httpd.conf dosyasından apachectl'ye geçme

Farklı Apache ayarlarını denemek için orjinal httpd.conf dosyasını düzenleyebilirsiniz. Eğer bir şeyler çalışmazsa, ayarları geri alabilirsiniz. Orjinal httpd.conf ile uğraşmak yerine yeni bir httpd.conf.debug kopyası oluşturup bu httpd.conf.debug dosyasını Apache ile aşağıdaki gibi -f seçeneği ile test amaçlı kullanabilirsiniz.

```
# apachectl -f conf/httpd.conf.debug
# httpd -k start -f conf/httpd.conf.debug
[Not: Yukarıdaki gibi apachectl ve httpd'nin ikisini de kullanabilirsiniz.]
# ps -ef | grep httpd
root 25080 1 0 23:26 00:00:00 /usr/sbin/httpd -f conf/httpd.conf.debug

apache 25099 25080 0 23:28 00:00:00 /usr/sbin/httpd -f conf/httpd.conf.debug
[Not: ps, httpd'yi httpd.conf.debug dosyası ile çalışıyor hâlde gösterdi.]
```

Değişikliklerden bir kere memnun kaldıysanız ve Apache, httpd.conf.debug ile herhangi bir problem olmadan çalışıyorsa, aşağıdaki gibi değişiklikleri httpd.conf dosyasına kopyalayıp Apache'yi normal olarak başlatabilirsiniz.

```
# cp httpd.conf.debug httpd.conf
# apachectl stop
# apachectl start
# ps -ef | grep httpd
root 25114 1 0 23:28 00:00:00 /usr/sbin/httpd -k start
daemon 25115 25114 0 23:28 00:00:00 /usr/sbin/httpd -k start
[Not: ps, httpd'nin varsayılan konfigürasyon dosyasıyla çalıştığını belirtiyor.]
```

İpucu 77. DocumentRoot'u httpd.conf dosyasını değiştirmeden geçici olarak kullanma

Websiteniz için farklı katman dendiğinizde ve varsayılan DocumentRoot altındaki orjinal dosyaları düzenlemek istemediğinizde, bu işlem çok kullanışlıdır.

Orjinal DocumentRoot dizininin (/var/www/html) bir kopyasını geçici bir DocumentRoot (/var/www/html_debug) dizininde oluşturun. Tüm ayarlamalarınızı bu geçici dizin altında yapın ve Apache'yi, aşağıda gösterildiği gibi -c seçeneği ile geçici dizin ile başlatın.

```
# httpd -k start -c "DocumentRoot /var/www/html_debug/"
```

Eğer varsayılan DocumentRoot (/var/www/html) kullanarak orjinal konfigürasyona dönmek istiyorsanız, aşağıdaki gibi basitçe Apache'yi yeniden başlatabilirsiniz.

```
# httpd -k stop  
# apachectl start
```

İpucu 78. Geçici olarak Log Level'i artırma

Bir hatayı ayıklarken, geçici olarak Apache'nin LogLevel'ini değiştirebilirsiniz, -e seçeneği kullanarak httpd.conf'u düzenlemeden LogLevel tâtimatı verebilirsiniz. Bu örnekte LogLevel, debug olarak ayarlanmıştır.

```
# httpd -k start -e debug  
[Sun Aug 17 13:53:06 2008] [debug] mod_so.c(246): loaded module  
auth_basic_module  
[Sun Aug 17 13:53:06 2008] [debug] mod_so.c(246): loaded module  
auth_digest_module
```

-e seçeneğine verebileceğiniz olası değerler:

- o debug
- o info
- o notice
- o warn
- o error
- o crit
- o alert
- o emerg

İpucu 79. Apache içindeki modülleri gösterme

Apache içerisinde derlenmiş modülleri gösterme

```
# httpd -l
Compiled in modules:
core.c
prefork.c
http_core.c
mod_so.c
```

Apache tarafından yüklenmiş statik ve dinamik modülleri gösterme

httpd'ye verilen -l seçeneği sadece statik modülleri gösterir. Oysa, -M seçeneği statik ve paylaşılan modüllerin hepsini gösterir.

```
# httpd -M
Loaded Modules:
core_module (static)
mpm_prefork_module (static)
http_module (static)
so_module (static)
auth_basic_module (shared)
auth_digest_module (shared)
authn_file_module (shared)
authn_alias_module (shared)
Syntax OK
```

İpucu 80. httpd.conf içerisinde kabul edilmiş şartları gösterme

Bu ipucu, tüm httpd.conf tâlimatlarını(şartları) gösteren genişletilmiş yardım olarak nitelendirilebilir. httpd.conf'un içerisinde kullanabileceğiniz tüm olası değerleri söyler. Özel bir Apache ayarı hakkında hızlıca bilgi almak için bu yöntem kullanışlıdır. (Ayarlar türkçeye çevrilmemiştir.)

```
# httpd -L

HostnameLookups (core.c)
"on" to enable, "off" to disable reverse DNS lookups, or "double" to
enable double-reverse DNS lookups
Allowed in *.conf anywhere

ServerLimit (prefork.c)
Maximum value of MaxClients for this run of Apache
Allowed in *.conf only outside <Directory>, <Files> or <Location>

KeepAlive (http_core.c)
Whether persistent connections should be On or Off
Allowed in *.conf only outside <Directory>, <Files> or <Location>

LoadModule (mod_so.c)
a module name and the name of a shared object file to load it from
Allowed in *.conf only outside <Directory>, <Files> or <Location>
```

İpucu 81. Değişiklik yaptıktan sonra httpd.conf'u doğrulamak

-t seçeneğini kullanarak Apache konfigürasyon dosyasında herhangi bir hata olup olmadığını kontrol edebilirsiniz. Aşağıda gösterilen örnekte, httpd.conf.debug dosyasının 148. satırında problem olduğu gösteriliyor.

```
# httpd -t -f conf/httpd.conf.debug
httpd: /etc/httpd/conf/httpd.conf.debug 148. satırında sözdizimi hatası:
/etc/httpd/modules/mod_auth_basicso sunucuya yüklenemedi:
/etc/httpd/modules/mod_auth_basicso: paylaşılmış nesne dosyası açılmıyor:
Böyle bir dosya ya da izin yok
```

Hatayı bir kez çözdükten sonra aynı komut "Syntax OK" çıktısı verecektir.

```
# httpd -t -f conf/httpd.conf.debug
Syntax OK
```

İpucu 82. httpd yapılandırma parametrelerini gösterme

-V (büyük v) seçeneği, Apache versiyon numarası ve Apache'yi yapılandırırken kullanılan tüm parametreleri gösterir.

```
# httpd -V
Sunucu versiyonu: Apache/2.2.9 (Unix)
Sunucu kurulumu: Tem 14 2008 15:36:56
Sunucunun Module Magic Number bilgisi: 20051115:15
Yüklenen sunucu: APR 1.2.12, APR-Util 1.2.12
Derlenirken kullanılanlar: APR 1.2.12, APR-Util 1.2.12
Mimâri: 32-bit
MPM Sunucu: Prefork
threaded: no
forked: yes (değişken işlem sayısı)
Sunucu şunlarla derlendi...
-D APACHE_MPM_DIR="server/mpm/prefork"
-D APR_HAS_SENDFILE
-D APR_HAS_MMAP
-D APR_HAVE_IPV6 (IPv4-mapped addresses enabled)
-D APR_USE_SYSVSEM_SERIALIZE
-D APR_USE_PTHREAD_SERIALIZE
-D SINGLE_LISTEN_UNSERIALIZED_ACCEPT
-D APR_HAS_OTHER_CHILD
-D AP_HAVE_RELIABLE_PIPED_LOGS
-D DYNAMIC_MODULE_LIMIT=128
-D HTTPD_ROOT="/etc/httpd"
-D SUEXEC_BIN="/usr/sbin/suexec"
-D DEFAULT_PIDLOG="logs/httpd.pid"
-D DEFAULT_SCOREBOARD="logs/apache_runtime_status"
-D DEFAULT_LOCKFILE="logs/accept.lock"
-D DEFAULT_ERRORLOG="logs/error_log"
-D AP_TYPES_CONFIG_FILE="conf/mime.types"
-D SERVER_CONFIG_FILE="conf/httpd.conf"
```

Eğer sadece Apache versiyon numarasını yazdırmak istiyorsanız, aşağıdaki gibi -v (küçük v) seçeneğini kullanmalısınız.

```
# httpd -v
Sunucu versiyonu: Apache/2.2.9 (Unix)
Sunucu kurulumu: Jul 14 2008 15:36:56
```

İpucu 83. Sadece istenildiğinde özel modülü yüklemek

Bazen, Apache'deki tüm modülleri yüklemek istemeyebilirsiniz. Örneğin, yalnızca LDAP (Basit Dizin Erişim Protokolü) testi yapmak istediğinizde Apache'ye ldap ile ilgili modülleri yüklemek istediniz. Aşağıdaki gibi yapmalısınız.

httpd.conf dosyasını düzenlemeli ve IfDefine şartını ekleyerek ldap'ı (load-ldap) çağırmanız gereklidir. (load-ldap yerine istediğiniz ismi verebilirsiniz.)

```
<IfDefine load-ldap>
LoadModule ldap_module modules/mod_ldap.so
LoadModule authnz_ldap_module modules/mod_authnz_ldap.so
</IfDefine>
```

ldap kullanarak test yaparken, ldap ile ilgili modülleri yüklemek istediğinizde, aşağıdaki gibi -D seçeneğine load-ldap vermeniz gerekir.

```
# httpd -k start -e debug -Dload-ldap -f /etc/httpd/conf/httpd.conf.debug
[Paz Ağu 17 14:14:58 2008] [debug] mod_so.c(246): ldap_module
modülü yüklendi.
[Paz Ağu 17 14:14:58 2008] [debug] mod_so.c(246): authnz_ldap_module
modülü yüklendi.
[Not:-D seçeneğine load-ldap verince, Apache'de ldap modülleri yüklendi.]

# apachectl start

[Not: ldap modüllerini yüklemek istemiyorsanız, Apache 'yi normal
olarak başlatır.]
```

Bölüm 11: Bash Kabuk Betiği

İpucu 84. `.bash_*` dosyalarını çalıştırma sırası

Aşağıdaki dosyaların çalıştırılma sırası nedir ?

- o `/etc/profile`
- o `~/.bash_profile`
- o `~/.bashrc`
- o `~/.bash_login`
- o `~/.profile`
- o `~/.bash_logout`

Oturum açılmış etkileşimli kabuk için çalıştırma sırası

Aşağıdaki sözde kod, bu dosyaların çalıştırılma sırasını açıklıyor.

```
çalıştır /etc/profile
IF ~/.bash_profile varsa THEN
    çalıştır ~/.bash_profile
ELSE
    IF ~/.bash_login varsa THEN
        çalıştır ~/.bash_login
    ELSE
        IF ~/.profile varsa THEN
            çalıştır ~/.profile
        END IF
    END IF
END IF
```

Etkileşimli kabuktan çıkarken, aşağıdaki çalıştırılma sırası uygulanır

```
IF ~/.bash_logout varsa THEN
    çalıştır ~/.bash_logout
END IF
```

`/etc/bashrc` dosyasının `~/.bashrc` dosyası tarafından çalıştırıldığını not alabilirsiniz.

```
# cat ~/.bashrc
if [ -f /etc/bashrc ]; then
    . /etc/bashrc
fi
```

Oturum açılmamış etkileşimli kabuk için çalıştırma sırası

```
IF ~/.bashrc varsa THEN
    çalıştır ~/.bashrc
END IF
```

Not: Etkileşimli olmayan kabuk başladığında, ENV ortam değişkeni kontrol edilir ve ENV değişkeninde tanımlı olan dosya çalıştırılır.

İpucu 85. Bash kabuğunda rastgele sayı üretmek

\$RANDOM adlı gömülü bash fonksiyonu ile 0 – 32767 arasında rastgele sayı oluşturabilirsiniz.

```
$ echo $RANDOM
22543

$ echo $RANDOM
25387

$ echo $RANDOM
647
```

İpucu 86. Kabuk betiğinde hata ayıklama

Kabuk betiğinin içinde en üstte “set -xv” kullanarak hata ayıklayabilirsiniz.

Hata ayıklama komutu olmadan kabuk betiği

```
$ cat filesize.sh
#!/bin/bash
for dosyaboyutu in $(ls -l . | grep "^-" | awk '{print $5}')
do
    let toplamboyut=$toplamboyut+$dosyaboyutu
done
echo "Geçerli dizindeki toplam dosya boyutu: $toplamboyut"
```

Çıktısı

```
$ ./filesize.sh
Geçerli dizindeki toplam dosya boyutu: 652
```


İçerisinde hata ayıklama komutu olan kabuk betiği

Aşağıdaki gibi kabuk betiğinin içine “set -xv” ekleyerek hata ayıklayacağımızı belirttik.

```
$ cat filesize.sh
#!/bin/bash
set -xv
for dosyaboyutu in $(ls -l . | grep "^-" | awk '{print $5}')
do
    let toplamboyut=$toplamboyut+$dosyaboyutu
done
echo "Geçerli dizindeki toplam dosya boyutu: $toplamboyut"
```

Çıktısı

```
$ ./fs.sh
++ ls -l .
++ grep '^-'
++ awk '{print $5}'
+ for dosyaboyutu in '$(ls -l . | grep "^-" | awk '\''{print $5}'\'' )'
+ let toplamboyut=+178
+ for dosyaboyutu in '$(ls -l . | grep "^-" | awk '\''{print $5}'\'' )'
+ let toplamboyut=178+285
+ for dosyaboyutu in '$(ls -l . | grep "^-" | awk '\''{print $5}'\'' )'
+ let toplamboyut=463+189
+ echo 'Geçerli dizindeki toplam dosya boyutu: 652'
Geçerli dizindeki toplam dosya boyutu: 652
```

Kabuk betiğini hata ayıklama seçeneğiyle çalıştırmak

Betiğin içinde set -xv yazmak yerine aşağıdaki gibi, betiği çalıştırırken hata ayıklama modunu belirtebilirsiniz.

```
$ bash -xv filesize.sh
```

İpucu 87. Kaçış işaretleri ve tırnaklar

Herhangi bir özel karakter kullanmadan echo ifadesi.

```
$ echo The Geek Stuff
The Geek Stuff
```

Echo ifadesi ile kullanılan noktalı virgül özel karakteri, komut sonlandırıcı görevi görür. Aşağıdaki örnekte, “The Geek” kelimesi echo ifadesi ile çalışır, “Stuff” kelimesi ise noktalı virgülden sonra geldiği için ayrı bir komut olarak algılanır.

```
$ echo The Geek; Stuff
The Geek
-bash: Stuff: Komut bulunamadı.
```

Noktalı virgölün önüne \ kaçış karakterini eklerseniz, özel anlamı olduğunu anlayacak ve noktalı virgölü ve sonrasını sorunsuz ekrana basacaktır.

```
$ echo The Geek\; Stuff
The Geek; Stuff
```

Tek Tırnak

Tırnakların içerisindeki her şeyin olduğu gibi yazdırılmasını istiyorsanız, tek tırnak kullanmalısınız. \$HOSTNAME gibi özel değişkenler, gerçek hostname değerini göstermek yerine \$HOSTNAME olarak yazdırılacaktır.

```
$ echo 'Bilgisayar adı=$HOSTNAME ; Geçerli kullanıcı=`whoami` ;
Mesaj=\$ , Dolar işaretidir'

Bilgisayar adı=$HOSTNAME ; Geçerli kullanıcı=`whoami` ;
Mesaj=\$ , Dolar işaretidir
```

Çift Tırnak

Özel değişkenlerin gerçek anlamlarını göstermek istiyorsanız çift tırnak kullanmalısınız.

```
$ echo "Bilgisayar adı=$HOSTNAME ; Geçerli kullanıcı=`whoami` ;
Message=\$ , Dolar işaretidir"

Bilgisayar adı=dev-db ; Geçerli kullanıcı=ramesh ;
Mesaj=$ , Dolar işaretidir
```

Çift tırnaklar ile örnek gösterimler aşağıda gösterilmiştir.

- o \$ Parametre belirteci
- o ` Ters tırnak
- o \ \$ Dolar işareti
- o \ ` Ters tırnak işareti
- o \" Çift tırnak işareti
- o \\ Ters slash (taksim) işareti

İpucu 88. Kabuk betiği içerisinde veri dosyası alanı okuma

Bu örnekte, kabuk betiği içerisinde dosyadan özel bir bölümü nasıl okuyacağınız ve onu nasıl yöneteceğiniz gösterilmiştir. Örneğin, varsayalım ki aşağıdaki gibi, işçiler.txt dosyası, {işçi-adı}:{işçi-numarası}:{bölüm-adı} şeklinde dilimlenmiş olsun.

```
$ cat işçiler.txt
Emma Thomas:100:Marketing
Alex Jason:200:Sales
Madison Randy:300:Product Development
Sanjay Gupta:400:Support
Nisha Singh:500:Sales
```

Aşağıdaki betik, bu dosyadan özel alanları nasıl okuyacağımızı açıklıyor.

```
$ vi oku.sh
#!/bin/bash
IFS=:
echo "İşçilerin adları:"
echo "-----"
while read isim numara bölüm
do
    echo "$isim , $bölüm bölümünde çalışıyor."
done < ~/işçiler.txt
```

Betiği çalıştırılabilir olarak ayarlayın ve çalıştırın.

```
$ chmod u+x oku.sh
$ ./oku.sh
İşçilerin adları:
-----
Emma Thomas , Marketing department bölümünde çalışıyor.
Alex Jason , Sales department bölümünde çalışıyor.
Madison Randy , Product Development department bölümünde çalışıyor.
Sanjay Gupta , Support department bölümünde çalışıyor.
Nisha Singh , Sales department bölümünde çalışıyor.
```

Bölüm 12: Sistem Denetleme ve Performans

İpucu 89. Free komutu

free komutu, fiziksel sistem (RAM) bilgisi ve takas alanı hafızası hakkında tüm gerekli bilgileri gösterir.

```
Söz dizimi: free [seçenekler]
```

Sistemimdeki toplam RAM ne kadardır ?

Aşağıdaki örnekteki sistemde toplam fiziksel hafıza 1 GB 'dır. Değerler KB cinsinden gösterilmiştir.

```
# free
      toplam    kullanılan    boş    paylaşımlı    tampon bellek    önbellek
Hfz:  1034624    1006696    27928           0           174136    615892
-/+ buffers/cache:    216668    817956
Takas: 2031608           0    2031608
```

RAM ve Takas Alanı içeren toplam hafıza ne kadardır ?

Aşağıdaki komut:

- o m seçeneği değerleri MB cinsinden göstermesini sağlar
- o t seçeneği fiziksel ve takas alanı hafızası değerlerinin toplamını gösterir.
- o o seçeneği yukarıdaki örnekteki buffers/cache satırını gizler.

```
# free -mto
      toplam  kullanılan  boş  paylaşımlı  tampon bellek  önbellek
Hfz:   1010      983      27      0      170      601
Takas: 1983      0      1983
Toplam: 2994      983      2011
```

İpucu 90. Top komutu

top komutu CPU, Hafıza Kullanımı, İşlemler listesi vb. çeşitli performans değerlerini gerçek zamanlı gösterir.

```
Söz dizimi: top [seçenekler]
```

CPU kullanımını içeren geçerli sistem durumunu nasıl görebilirim ?

Komut satırından top komutunu hiçbir seçenek olmadan çalıştırdığınızda aşağıdaki gibi bir ekranla karşılaşacaksınız. Top komutu, siz “ctrl+c” yapmadıkça veya çıkış için q’ya basmadıkça gerçek zamanlı değerler gösterir.

```
# top
top - 13:10:13 up 171 days, 20:21, 3 users, load average: 0.01, 0.05, 0.00
Tasks: 194 total, 1 running, 193 sleeping, 0 stopped, 0 zombie
Cpu(s): 0.6% us, 0.7% sy, 0.0% ni, 98.7% id, 0.0% wa, 0.0% hi, 0.0% si
Mem: 1034624k total, 1007420k used, 27204k free, 174540k buffers
Swap: 2031608k total, 0k used, 2031608k free, 615904k cached
PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
11912 apache 15 0 31828 13m 3916 S 1 0.2 0:46.35 httpd
19299 oracle 19 0 279m 18m 17m S 1 0.2 0:00.03 oracle
11398 jsmith 16 0 107m 28m 6404 S 0 0.4 0:03.07 perl
```

Yukarıdaki top komutunun çıktısı nasıl okunur ?

- o 1. satırdaki “top”, sistemin 171 gündür çalışır durumda olduğunu belirtiyor.
- o 2. satırdaki “Tasks”, toplam çalışan, uyuyan, duran ve zombi işlem sayısını gösterir.
- o 3. satırdaki “Cpu(s)”, sistemin şu andaki CPU kullanımını gösterir. Örnekte, CPU’nun %98.7’si boşta.
- o 4. satırdaki “Mem” ve 5. satırdaki “Swap” hafıza bilgisini gösterir. Free komutuyla aynı bilgileri verir.
- o Satırların geri kalanı, varsayılan olarak CPU kullanımına göre sıralanmış sistemdeki tüm aktif işlemleri (process) gösterir. En fazla CPU kullanan işlem en üstte gösterilir.

Top komutu'na uygun birkaç komut satırı ve etkileşimli seçenek var. Hadi top komutu için temel seçenekleri inceleyelim.

En fazla hafıza kullanan işlemler nasıl belirlenir ?

Top komutunun çıktısı gösterilirken F'e bastığınızda aşağıdaki mesajı gösterir ve n ardından enter'a bastığınızda sıralanabilir tüm alanları gösterir. Bu, hafıza kullanımına göre sıralanmış çıktıyı görüntüler.

“Pencere 1:Def için geçerli sıralı alan:K

Geri dönmek için herhangi bir tuşa basınız.”

Top komutu çıktısına CPU zamanı gibi ek alanlar nasıl eklenir ?

Top komutu çalışırken f'e bastığınızda aşağıdaki mesajı gösterir, l'ye bastığınızda tüm CPU zamanı gösterilen kolonlara ekleyerek tüm alanları gösterir.

“Pencere 1:Def için Geçerli alanlar: AEHIOQTWKNMbcdfgjplrsuvyzX

Geri dönmek için herhangi bir tuşa basınız.”

Çalışan işlemlerin tam yolu ve parametreleri nasıl alınır ?

Top komutu çalışırken c'ye bastığınızda çalışan işlemin tam yolunu gösterecektir.

Aşağıdaki örnekte httpd yerine /usr/local/apache2/bin/httpd gösterilmiştir.

```
PID  USER  PR NI VIRT RES SHR  S   %CPU %MEM TIME+ COMMAND
11912  apache 15 0 31   828 13m  3916 S 1 0.2 0:46.35 /usr/local/apache2/bin/httpd
```

Top komutunda CPU'ların durumu nasıl görüntülenir ?

Top komutu çalışırken 1 'e bastığınızda aşağıdaki gibi, makinedeki bireysel CPU'ların performans verilerini gösterecektir.

```
top - 13:10:13 up 171 days, 20:21, 3 users, load average: 0.01, 0.05, 0.00
Tasks: 194 total, 1 running, 193 sleeping, 0 stopped, 0 zombie
Cpu0  : 10.2% us, 2.6% sy, 0.0% ni, 86.8% id, 0.3% wa, 0.0% hi, 0.0% si
Cpu1  : 9.6% us, 8.0% sy, 0.0% ni, 82.4% id, 0.0% wa, 0.0% hi, 0.0% si
Cpu2  : 1.3% us, 1.3% sy, 0.0% ni, 95.0% id, 2.3% wa, 0.0% hi, 0.0% si
Cpu3  : 0.0% us, 0.0% sy, 0.0% ni, 100.0% id, 0.0% wa, 0.0% hi, 0.0% si
Mem: 1034624k total, 1007420k used, 27204k free, 174540k buffers
Swap: 2031608k total, 0k used, 2031608k free, 615904k cached
```

İpucu 91. Ps komutu

ps komutu (process status: işlem durumu) , anlık olarak tüm aktif işlemleri gösterir.

```
Söz dizimi : ps [seçenekler]
```

Sistemde çalışan tüm işlemler nasıl görüntülenir ?

Aşağıdaki gibi "ps aux" kullanarak, sistemdeki tüm işlemleri görüntüleyebilirsiniz.

```
# ps aux | more
USER  PID  %CPU %MEM VSZ  RSS  TTY STAT START TIME COMMAND
root   1    0.0  0.0 2044  588  ?   Ss   Jun27 0:00 init [5]
apache 31186 0.0  1.6 23736 17556 ?    S   Jul26 0:40 /usr/local/apache2/bin/httpd
apache 31187 0.0  1.3 20640 14444 ?    S   Jul26 0:37 /usr/local/apache2/bin/httpd
```

Ayrıca "ps -ef | more" komutu da aşağı yukarı aynı çıktıyı verir.

İşlem ağacını yazdırma

İşlem listesini ağaç formatında göstermek için “ps axuf” veya “ps -ejH” kullanabilirsiniz. Ağaç yapısı, işlemi ve onun ebeveyn işlemini hızlıca görselleştirmeye yardımcı olur. Sadelik açısından daha iyidir.

```
# ps axuf
root  Oct14  0:00  /opt/VRTSralus/bin/beremote
root  Oct14  0:00  \_ /opt/VRTSralus/bin/beremote
root  Oct14  0:00  \_ /opt/VRTSralus/bin/beremote
root  Oct14  0:00  \_ /opt/VRTSralus/bin/beremote
root  Oct14  0:01  \_ /opt/VRTSralus/bin/beremote
root  Oct14  0:00  \_ /opt/VRTSralus/bin/beremote
root  Dec03  0:01  /usr/local/sbin/sshd
root  Dec22  1:08  /usr/local/sbin/sshd
root  23:35  0:00  \_ /usr/local/sbin/sshd
511  23:35  0:00  \_ -bash
511  \_ ps axuf
```

Not: Ayrıca pstree komutunu kullanarak işlem listesini ağaç yapısında gösterebilirsiniz.

Bir kullanıcının sahip olduğu işlemleri gösterme

Aşağıdaki komut Linux kullanıcı adı “oracle” olan kullanıcının sahip olduğu tüm işlemleri gösterir.

```
$ ps U oracle
PID  TTY  STAT  TIME  COMMAND
5014  ?    Ss    0:01  /oracle/bin/tnslsnr
7124  ?    Ss    0:00  ora_q002_med
8206  ?    Ss    0:00  ora_cjq0_med
8852  ?    Ss    0:01  ora_pmon_med
8854  ?    Ss    0:00  ora_psp0_med
8911  ?    Ss    0:02  oraclemed (LOCAL=NO)
```


Geçerli kullanıcının sahip olduğu işlemleri gösterme

Aşağıdaki komut geçerli kullanıcının sahip olduğu tüm işlemleri gösterir.

```
$ ps U $USER
PID  TTY  STAT TIME  COMMAND
10329 ?    S    0:00  sshd: ramesh@pts/1,pts/2
10330 pts/1 Ss   0:00  -bash
10354 pts/2 Ss+  0:00  -bash
10530 pts/1 R+   0:00  ps U ramesh
```

İpucu 92. Df komutu

df komutu (disk free: boş disk), bölünmüş dosya sisteminde uygun olan boş disk alanının toplam miktarını gösterir.

```
Söz dizimi: df [seçenekler] [isim]
```

Sistemde kaç GB boş alan olduğu nasıl öğrenilir ?

-h seçeneği, değerleri okunabilir formatta (Kb için K, Mb için M ve Gb için G) gösterir. Aşağıdaki örnek çıktıda, / dosya sistemi (kök dizin) 17 GB boş alana, /home/user ise 70 GB boş alana sahiptir.

```
# df -h
Dosyasistemi  Boyut  Kullanılan  Uygun  Kullanım%  Mount edilmiş
/dev/sda1     64G    44G        17G    73%        /
/dev/sdb1     137G   67G        70G    49%        /home/user
```

Sistemde hangi dosya sistemi tipine sahip olunduğu nasıl öğrenilir ?

-T seçeneği, dosya sistemi tipleri hakkında bilgi verir. Aşağıdaki örnekte / ve /home/user dosyasistemleri ext2 tipindedir. -a seçeneği, tüm dosya sistemlerini gösterir.

```
# df -Tha
```

Dosyasistemi	Tip	Boyut	Kullanılan	Uygun	Kullanım%	Mount edilmiş
/dev/sda1	ext2	64G	44G	17G	73%	/
/dev/sdb1	ext2	137G	67G	70G	49%	/home/user
none	proc	0	0	0	-	/proc
none	sysfs	0	0	0	-	/sys
none	devpts	0	0	0	-	/dev/pts
none	tmpfs	2.0G	0	2.0G	0%	/dev/shm

İpucu 93. Kill komutu

kill komutu, çalışan bir işlemi bitirmek için kullanılır. Bu komut genel olarak durdurulmuş veya yanıt vermeyen işlemleri bitirmek için kullanılır.

```
Söz dizimi: kill [seçenekler] [pids(process id:işlem numarası)| komutlar]
```

Durdurulmuş bir işlem nasıl sonlandırılır ?

Öncelikle, ps komutu kullanarak işlemin process id (işlem numarası) 'nı belirleyin.

İşlem numarasını bir kez bilerseniz, kill komutuna parametre olarak verebilirsiniz.

Aşağıdaki örnekte durdurulmuş apache httpd işleminin nasıl sonlandırılacağı gösterilmiştir. Apache'yi durdurmak için "apachectl stop" yazmanız yeterlidir.

```
# ps aux | grep httpd
```

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
apache	31186	0.0	1.6	23736	17556	?	S	Jul26	0:40	/usr/local/apache2/bin/httpd
apache	31187	0.0	1.3	20640	14444	?	S	Jul26	0:37	/usr/local/apache2/bin/httpd

```
# kill 31186 31187
```

Yukarıdaki komut, SIGTERM adında bir sinyal göndererek işlemi nazikçe sonlandırmayı deniyor. Eğer işlem sonlanmazsa, -9 seçeneğini kullanarak SIGKILL adında sinyal göndererek işlemi zorla durdurabilirsiniz. Bir işlemi sonlandırabilmek için ya o işlemin sahibi olmanız ya da yetkili bir kullanıcı olmanız gerekir.

```
# kill -9 31186 31187
```

Çoklu işlemleri kolaylıkla sonlandırmak için bir diğer yol da .bash_profile dosyasına iki fonksiyon eklemektir.

```
function islemara ()
{
    ps aux | grep "$1" | grep -v 'grep'
}
function islemsonlandir ()
{
    [ $# -eq 0 ] && echo "usage: $FUNCNAME STRING" && return 0
    local pid
    pid=$(ps ax | grep "$1" | grep -v grep | awk '{ print $1 }')
    echo -e "Sonlandırılıyor '$1' / işlem(ler):\n$pid"
    kill -SIGTERM $pid
}
```

Şimdi, bir httpd işlemini belirleyelim ve sonlandıralım.

```
# islemara http
USER  PID  %CPU %MEM VSZ  RSS  TTY STAT START TIME COMMAND
apache 31186 0.0 1.6 23736 17556 ? S Jul26 0:40 /usr/local/apache2/bin/httpd
apache 31187 0.0 1.3 20640 14444 ? S Jul26 0:37 /usr/local/apache2/bin/httpd

# islemsonlandir httpd
Sonlandırılıyor 'httpd' / işlem(ler): 31186 31187
```

İpucu 94. Du komutu

du komutu (disk usage: disk kullanımı), özel bir dizin ve onun alt dizinleri için kullanılabilir boş alanı yazdırır.

Ev dizini ve altdizinlerinde kullanılabilir boş alanlar nasıl öğrenilir ?

Aşağıdaki örnekte, -s seçeneği summary (özet) anlamına gelir ve sadece /home/jsmith 'in ve tüm altdizinlerinin toplam boyutunu gösterir. -h seçeneği, çıktıyı okunabilir (KB yerine K, MB için M ve GB için G) formatta verir. ~ (tilde) kullanıcı ev dizinini belirtir. Bu komut “du -sh /home/jsmith” ile aynıdır.

```
# du -sh ~  
320M /home/jsmith
```

Yukarıdaki komutu -s seçeneği olmadan çalıştırırsanız, /home/jsmith altındaki tüm altdizinleri listeler.

İpucu 95. lsof komutları

Lsof komutu, sistemdeki tüm açık dosyaları listeleyen, ls open files (ls açık dosya)'nın kısaltmasıdır. Açık dosyalar, ağ bağlantıları, cihazlar ve dizinleri içerir. Lsof komutunun çıktısı aşağıdaki kolonlara sahip olacaktır.

- o COMMAND => işlem adı
- o PID => işlem numarası (process id)
- o USER => kullanıcı adı
- o FD => dosya tanımlayıcı (file descriptor)
- o TYPE => dosyanın node tipi
- o DEVICE => cihaz numarası
- o SIZE => dosya boyutu
- o NODE => node numarası
- o NAME => dosya adının tam yolu

Sistemin tüm açık dosyalarını görüntüleme

Aşağıdaki gibi lsof komutunu hiçbir parametre olmadan çalıştırabilirsiniz.

```
# lsof | more
COMMAND PID USER FD TYPE DEVICE SIZE NODE NAME
init 1 root cwd DIR 8,1 4096 2 /
init 1 root rtd DIR 8,1 4096 2 /
init 1 root txt REG 8,1 32684 983101 /sbin/init
init 1 root mem REG 8,1 106397 166798 /lib/ld-2.3.4.so
init 1 root mem REG 8,1 1454802 166799 /lib/tls/libc-2.3.4.so
init 1 root mem REG 8,1 53736 163964 /lib/libsepol.so.1
init 1 root mem REG 8,1 56328 166811 /lib/libselinux.so.1
init 1 root 10u FIFO 0,13 972 /dev/initctl
migration 2 root cwd DIR 8,1 4096 2 /
...
```

Kullanıcının açık dosyalarını görme

-u seçeneği, belirtilen kullanıcının tüm açık dosyalarını göstermek için kullanılır.

```
# lsof -u ramesh
vi 7190 ramesh txt REG 8,1 474608 475196 /bin/vi
sshd 7163 ramesh 3u IPv6 15088263 TCP dev-db:ssh->abc-12-12-12-12.socal.res.rr.com:2631 (ESTABLISHED)
```

Bu komut, sistem yöneticisinin hangi kullanıcının sistemde neler çalıştırdığı hakkında fikir sahibi olmasını sağlar.

Dosyanın kullanıcılarını listeleme

Özel bir dosyayı kullanan tüm kullanıcıları görmek istiyorsanız, lsof 'u aşağıdaki gibi kullanmalısınız. Örnekte, şu anda vi kullanan tüm kullanıcıları gösteriyor.

```
# lsof /bin/vi
COMMAND PID USER FD TYPE DEVICE SIZE NODE NAME
vi 7258 root txt REG 8,1 474608 475196 /bin/vi
vi 7300 ramesh txt REG 8,1 474608 475196 /bin/vi
```

İpucu 96. Sar komutu

Sar komutu, sysstat paketiyle gelir. Sysstat'ın yüklü olduğuna emin olun. Eğer sisteminizde sar yüklü değilse, [buradan](#) sysstat'ı indirebilirsiniz.

Sar, CPU, hafıza, girdi/çıkıtı, bağlantılar, kesmeler vb. dahil tüm sistem kaynaklarını gösteren mükemmel bir gözlem aracıdır.

Sar performans verilerini toplar, raporlar (gösterir) ve kaydeder. Hadi üç ayrı yönünü inceleyelim.

Sadc – Sistem etkinliği veri toplayıcı (System activity data collector)

/usr/lib/sadc komutu belirli zaman aralığında sistem verilerini toplar. Bu işlem, /var/log/sa/sa[gg] (gg, günü belirtir) altında bulunan günlük etkinliği kullanır.

Sa1 kabuk betiği

/usr/lib/sa1 , sırayla /usr/lib/sadcs 'i çağırır. Sa1 aşağıda gösterildiği gibi crontab tarafından da çağırılabilir. İhtiyacınıza bağlı olarak her 5 veya 15 dakikada bir çalıştırılır. Ben, aşağıdaki gibi crontab'da her 5 dakikada bir planlamayı tercih ediyorum.

```
*/5 * * * * root /usr/lib/sa/sa1 1 1
```

Sa2 kabuk betiği

/usr/lib/sa2 , /var/log/sa/sa[gg] dosyasına günlük raporları yazan betiktir. Sa2, gece yarısında günde bir kez olmak üzere crontab tarafından çağırılır.

```
# 59 23 * * * root /usr/lib/sa/sa2 -A
```

Not: /etc/cron.d/sysstat dosyası, birkaç varsayılan sa1 ve sa2 değerleri içeren sysstat paketiyle gelir. Bunu istediğiniz gibi değiştirebilirsiniz.

Sar komutu kullanarak CPU istatistiklerini gösterme

```
# sar -u
Linux 2.6.9-42.ELsmp (dev-db) 01/01/2009
12:00:01 AM CPU %user %nice %system %iowait %idle
12:05:01 AM all 3.70 0.00 0.85 0.00 95.45
12:10:01 AM all 4.59 0.00 1.19 0.06 94.16
12:15:01 AM all 3.90 0.00 0.95 0.04 95.11
12:20:01 AM all 4.06 0.00 1.00 0.01 94.93
12:25:01 AM all 3.89 0.00 0.87 0.00 95.23
12:30:01 AM all 3.89 0.00 0.87 0.00 95.23
.....
Average:      all 4.56 0.00 1.00 0.15 94.29
```

Not: Tek tek CPU'lar için performans verisine ihtiyacınız varsa, aşağıdaki komutu çalıştırabilirsiniz.

```
# sar -u -P ALL
```

Sar komutu kullanarak disk girdi/çıkışı istatistiklerini gösterme

```
# sar -d
Linux 2.6.9-42.ELsmp (dev-db) 01/01/2009
12:00:01 AM DEV tps rd_sec/s wr_sec/s
12:05:01 AM dev2-0 1.65 1.28 45.43
12:10:01 AM dev8-1 4.08 8.11 21.81
.....
Average:      dev2-0 4.66 120.77 69.45
Average:      dev8-1 1.89 3.17 8.02
```

Sar komutu kullanarak bağlantı istatistiklerini gösterme

```
# sar -n DEV | more
Linux 2.6.9-42.ELsmp (dev-db) 01/01/2009
12:00:01 AM IFACE rxpck/s txpck/s rxbyt/s txbyt/s rxcmp/s txcmp/s rxmcast/s
12:05:01 AM lo 0.17 0.16 25.31 23.33 0.00 0.00 0.00
12:10:01 AM eth0 52.92 53.64 10169.74 12178.57 0.00 0.00 0.00

# sar -n SOCK |more
Linux 2.6.9-42.ELsmp (dev-db) 01/01/2009
12:00:01 AM totsck tcpsck udpsck rawsck ip-frag
12:05:01 AM 50 13 3 0 0
12:10:01 AM 50 13 4 0 0
12:15:01 AM 53 13 5 0 0
```

İpucu 97. Vmstat komutu

Tipik bir performans gözleme için tek ihtiyacınız vmstat komutudur. Bu komut hafıza, takas, girdi/çıkıtı, sistem ve CPU performans bilgisini gösterir.

Aşağıdaki komut, vmstat'ı her 1 saniyede bir toplamda 100 kez çalıştıracaktır.

```
# vmstat 1 100
procs -----memory----- --swap--  --io--  --system--  ----cpu-----
 r b  swpd free  buff  cache    si so    bi bo    in  cs  us sy id  wa
 0 0   0  282120 134108 5797012  0 0     0 2    0  0   0 0 100 0
 0 0   0  282120 134108 5797012  0 0     0 0   1007 359 0 0 100 0
 0 0   0  282120 134108 5797012  0 0     0 0   1117 577 0 0 100 0
 0 0   0  282120 134108 5797012  0 0     0 0   1007 366 0 0 100 0
```

İşlemler bölümü (procs)

- o r alanı: Toplam çalıştırılabilir işlem sayısı (runnable)
- o b field: Toplam bloklanmış işlem sayısı (blocked)

Hafıza bölümü (memory)

- o Swpd alanı: Kullanılan takas alanı
- o Free alanı: Kullanılabilir RAM miktarı
- o Buff alanı: Arabellek tarafından kullanılan RAM miktarı (buffer)
- o Cache alanı: Dosya sistemi önbelleği tarafından kullanılan RAM miktarı

Takas bölümü (swap)

- o Si alanı: Diskten saniyede takas alınan hafıza miktarı (swap in)
- o So alanı: Diske saniyede takas verilen hafıza miktarı (swap out)

Girdi/Çıktı bölümü (IO)

- o Bi alanı: Diskten alınan bloklar (block in)
- o Bo alanı: Diske gönderilen bloklar (block out)

Sistem bölümü (system)

- o In alanı: Saniyedeki kesme sayısı (interrupt)
- o Cs alanı: Saniyedeki içerik değiştirme sayısı (context switching)

CPU bölümü

- o Us alanı: Kullanıcı kodu çalıştırırken geçen süre (user code)
- o Sy alanı: Çekirdek kodu çalıştırırken geçen süre
- o Id alanı: Boş geçen süre
- o Wa alanı: Girdi/çıkı için beklerken geçen süre

İpucu 98. Netstat komutu

Netstat komutu ağ bağlantıları, yönlendirme tabloları, arayüz istatistikleri gibi ağ ile ilgili bilgileri gösterir. Aşağıdaki birkaç örnek, netstat'ın nasıl kullanıldığını gösterir.

Netstat kullanarak aktif internet bağlantılarını ve soketleri gösterme

```
# netstat -an
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address Foreign Address State
tcp 0 0 0.0.0.0:5666 0.0.0.0:* LISTEN
tcp 0 0 0.0.0.0:111 0.0.0.0:* LISTEN
tcp 0 0 0.0.0.0:4086 0.0.0.0:* LISTEN
.....

Active UNIX domain sockets (servers and established)
Proto RefCnt Flags Type State I-Node Path
unix 2 [ ACC ] STREAM LISTENING 7894 /tmp/.font-unix/fs7100
unix 2 [ ACC ] STREAM LISTENING 9662 /tmp/.gdm_socket
unix 2 [ ACC ] STREAM LISTENING 10897 @/tmp/fam-root-
```

İşlem numarası ve program adı ile aktif bağlantıları gösterme

Bu, her programın kendine özgü kullandığı ağ bağlantılarını belirlemede yardımcı olur.

```
# netstat -tap
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Prg name
tcp    0      0 *:nrpe                  *.*                     LISTEN      16277/xinetd
tcp    0      0 localhost.localdomain:smtp *.*                     LISTEN      7263/sendmail
tcp    34     0 localhost.localdomain:54221 localhost.localdomain:4089 CLOSE_WAIT 29881/httpd
tcp    0      3216 dev-db:ssh              cpe-76-94-215-154.soca:4682ESTABLISHED11717/sshd
```

Yönlendirme tablosunu gösterme

```
# netstat --route
Kernel IP routing table
Destination Gateway      Genmask      Flags MSS Window  irtt Iface
192.168.1.0 *            255.255.255.0 U        0     0      0 eth0
162.244.0.0 *            255.255.0.0  U        0     0      0 eth0
default     192.168.1.1 0.0.0.0      UG       0     0      0 eth0
```

Ham ağ istatistiklerini gösterme

```
# netstat --statistics --raw
Ip:
    11080343 total packets received
    0 forwarded
    1 with unknown protocol
    0 incoming packets discarded
    11037744 incoming packets delivered
    11199763 requests sent out
Icmp:
    577135 ICMP messages received
    64 input ICMP message failed.
    ICMP input histogram:
        destination unreachable: 537
        timeout in transit: 65
        source quenches: 2
        echo requests: 576476
        echo replies: 12
        timestamp request: 3
        address mask request: 3
    581558 ICMP messages sent
    0 ICMP messages failed
    ICMP output histogram:
        destination unreachable: 5079
        echo replies: 576476
        timestamp replies: 3
```

Çeşitli Netstat komutları

- o # netstat --tcp --numeric : Makineye giren ve çıkan TCP bağlantılarını listeler.
- o # netstat --tcp --listening --programs : Sunucunun, programların dinlediği belirli TCP portlarını gösterir.
- o # netstat -rnC : Yönlendirme önbelleğini gösterir.

İpucu 99. Sysctl komutu

Sysctl komutu kullanarak, Linux çekirdek parametreleri anında değiştirilebilir. Sysctl Linux çekirdek parametrelerini çalışma esnasında yapılandırmaya yardım eder.

```
# sysctl -a
dev.cdrom.autoclose = 1
fs.quota.writes = 0
kernel.ctrl-alt-del = 0
kernel.domainname = (none)
kernel.exec-shield = 1
net.core.somaxconn = 128
net.ipv4.tcp_window_scaling = 1
net.ipv4.tcp_wmem = 4096 16384 131072
net.ipv6.route.mtu_expires = 600
sunrpc.udp_slot_table_entries = 16
vm.block_dump = 0
```

/etc/sysctl.conf'taki çekirdek parametresini kalıcı olarak düzenleme

/etc/sysctl.conf'taki çekirdek parametresini düzenledikten sonra, sysctl -p komutunu çalıştırarak değişiklikleri işleyebilirsiniz. Değişiklikler, yeniden başlatmadan sonra da aynı kalacaktır.

```
# vi /etc/sysctl.conf
# sysctl -p
```

Çekirdek parametresini geçici olarak düzenleme

Bir çekirdek parametresini geçici olarak düzenlemek için, aşağıdaki komutu çalıştırın. Yeniden başlatmadan sonra ayarlar kaybolacaktır.

```
# sysctl -w {değişken-adi=değer}
```

İpucu 100. Nice komutu

Çekirdek, en iyi değeri bazlı (nice değeri) bir işlem için gerekli olan işlemci zamanını ölçer. Olası nice değerleri için aralık -20 ile 20 arasındadır. Nice değeri -20 olan işlem en yüksek önceliğe sahipken, nice değeri 20 olan işlem ise en düşük önceliğe sahiptir.

Tüm işlemlerin nice değerlerini göstermek için aşağıdaki gibi ps axl kullanın.

```
# ps axl
F UID PID PPID PRI NI VSZ RSS WCHAN STAT TTY TIME COMMAND
4 0 1 0 16 0 2172 552 - S ? 0:17 init [5]
1 0 3 1 34 19 0 0 ksofti SN ? 3:18 [ksoftirqd/0]
1 0 10 1 5 -10 0 0 worker S< ? 0:01 [events/0]
4 0 5145 1 25 10 32124 18592 - Sns ? 0:08 /usr/bin/python /usr/bin/rhn-applet-
gui --sm-client-id default4
4 0 5147 5142 16 0 3528 604 - S ? 0:00 /sbin/pam_timestamp_check -d root
1 503 17552 4180 16 0 14208 3920 - S ? 0:01 /home/www/apache2/bin/httpd -f
/home/www/apache2/conf/httpd.conf -k start
```

Bir kabuk betiğine nasıl en düşük önceliği atayabiliriz? (yüksek nice değeri)

Aşağıdaki örnekte, nice-test.sh betiğini arkaplanda çalıştırdığımızda, nice değeri 0'ı gösterir.

```
$ ./nice-test.sh &
[3] 13009
$ ps axl | grep nice-test
0 509 13009 12863 17 0 4652 972 wait S pts/1 0:00 /bin/bash ./nice-
test.sh
[Not: 6. kolon nice değerinin 0 olduğunu gösteriyor.]
```

Şimdi, aynı kabuk betiğini aşağıdaki gibi farklı nice değeriyle çalıştıralım.

```
$ nice -10 ./nice-test.sh &
[1] 13016
$ ps axl | grep nice-test
0 509 13016 12863 30 10 4236 968 wait SN pts/1 0:00 /bin/bash ./nice-test.sh
[Not: 6. kolondaki 10 değeri, kabuk betiği için nice değeridir.]
```

Bir kabuk betiğine nasıl en yüksek öncelik atayabiliriz? (düşük nice değeri)

Aşağıdaki örnekte, nice-test.sh betiğine -10 (eksi 10) nice değeri atandı.

```
$ nice --10 ./nice-test.sh &
[1] 13021
$ nice: öncelik ayarlanmadı: İzin reddedildi.
```

Not: Sadece kök (root) kullanıcı eksi nice değeri atayabilir. Kök kullanıcı olarak giriş yapın ve aynısını deneyin. Nice komutunda, 10'dan önce iki tire olduğunu unutmayın.

```
# nice --10 ./nice-test.sh &
[1] 13060
# ps axl | grep nice-test
4 0 13060 13024 10 -10 5388 964 wait S< pts/1 0:00 /bin/bash ./nice-test.sh
[Not: 6. kolondaki -10 değeri, kabuk betiğinin nice değeridir.]
```

İpucu 101. Renice komutu

Renice, çalışan bir işlemin zamanlama önceliğini değiştirir.

Çalışan bir işlemin önceliği nasıl azaltılır? (nice değerini artırmak)

Aşağıdaki örnekte, mevcut kabuk betiği 10 nice değeri ile çalışıyor. (ps çıktısının 6. kolonu)

```
$ ps axl | grep nice-test
0 509 13245 13216 30 10 5244 968 wait SN pts/1 0:00 /bin/bash ./nice-test.sh
```

Nice değerini artırmak (önceliği azaltmak) için renice komutunu aşağıdaki gibi çalıştırabilirsiniz.

```
$ renice 16 -p 13245
13245: eski öncelik 10, yeni öncelik 16
$ ps axl | grep nice-test
0 509 13245 13216 36 16 5244 968 wait SN pts/1 0:00 /bin/bash ./nice-test.sh

[Not: Şimdi, 6. kolondaki nice-test.sh'nin (işlem numarası 13245) nice değeri 16'yı gösteriyor.]
```

Çalışan bir işlemin önceliği nasıl artırılır? (nice değerini azaltmak)

Aşağıdaki örnekte, mevcut kabuk betiği 10 nice değeri ile çalışıyor. (ps çıktısının 6. kolonu)

```
$ ps axl | grep nice-test
0 509 13254 13216 30 10 4412 968 wait SN pts/1 0:00 /bin/bash ./nice-test.sh
```

Artan öncelikte, aşağıdaki gibi daha düşük nice değeri vermelisiniz. Ancak, sadece kök kullanıcı, çalışan bir işlemin önceliğini artırabilir, aksi takdirde aşağıdaki gibi hata mesajı alırsınız.

```
$ renice 5 -p 13254
renice: 13254: setpriority: Permission denied
Login as root to increase the priority of a running process
$ su -
# renice 5 -p 13254
13254: old priority 10, new priority 5
# ps axl | grep nice-test
0 509 13254 13216 25 5 4412 968 wait SN pts/1 0:00 /bin/bash ./nice-test.sh

[Not: En başta izin hatası verdi.Sonra root olarak giriş yapıldı ve daha düşük nice değeri verildi.(öncelik artırıldı)]
```