

6.Doküman

Bilgi Alma Komutları

Konu Etiketleri

bilgi alma , uname , hostname , lsb_release , whoami , who , w , uptime , date , cal , which , whereis , locate , dmidecode , fdisk , df , du , free , modinfo , stat , vmstat , history

Bilgi Almak

Bilgi alma komutları sistemimizde bulunan herhangi bir şey hakkında(programlar, dosyalar, kayıtlar, ayarlar vb..) bilgi almamızı sağlayan komutlara verilen genel isimdir ve bu komutlara gerekli durumlarda çok sık başvururuz. Şimdi bilgi alma komutlarının neler olduğuna ve kullanım şekillerine bakalım.

uname

Tek başına çekirdek adını verse de aldığı parametreler ile farklı bilgiler de sunabilir. Kısaca bilmemiz gereken; **bu komut genel olarak sistemde kullanılan çekirdek hakkında bilgiler verir**. Hangi parametrenin ne iş yaptığını **man uname** komutu ile öğrenebileceğinizi biliyorsunuz. Bizde kullanım şekillerini **man** kılavuz sayfası üzerinden açıklayalım.

```
└─$ uname
Linux
```

Eğer parametresiz kullanıldıysa **-s** parametresi gibi **yalnızca çekirdek ismini** gösterir. Yani **uname** komutu ile **uname -s** komutu aynıdır. Parametreler ve ifade ettikleri;

-a, --all Bu parametre ile bütün bilgileri aşağıdaki seçenek sırasıyla gösterilir; **-p** ve **-i** bilinmiyorsa atlanır.

```
└─$ uname -a
Linux taylan 5.9.0-kali1-amd64 #1 SMP Debian 5.9.1-1kali2 (2020-10-29) x86_64 GNU/Linux
```

-s, -- çekirdek ismini gösterir.

```
└─$ uname -s
Linux
```

-n, --nodename ağ düğümü konak ismini diğer bir deyişle hostname(host adını) gösterir.

```
└─$ uname -n
taylan
```

-r, --kernel-release çekirdeğin derleniş sürümünü gösterir.

```
└─$ uname -r
5.9.0-kali1-amd64
```

-v, --kernel-version çekirdeğin sürümünü gösterir.

```
└─$ uname -v
#1 SMP Debian 5.9.1-1kali2 (2020-10-29)
```

-m, --machine makine donanım ismini gösterir.

```
└─$ uname -m
x86_64
```

-p, --processor işlemci türünü veya "unknown" gösterir.

```
└─$ uname -p
unknown
```

-i, --hardware-platform donanım platformunu veya "unknown" gösterir.

```
└─$ uname -i
unknown
```

-o, --operating-system işletim sistemini gösterir.

```
└─$ uname -o
GNU/Linux
```

--help Bu yardım iletisini gösterir ve çıkar. --version Sürüm bilgilerini gösterir ve çıkar.

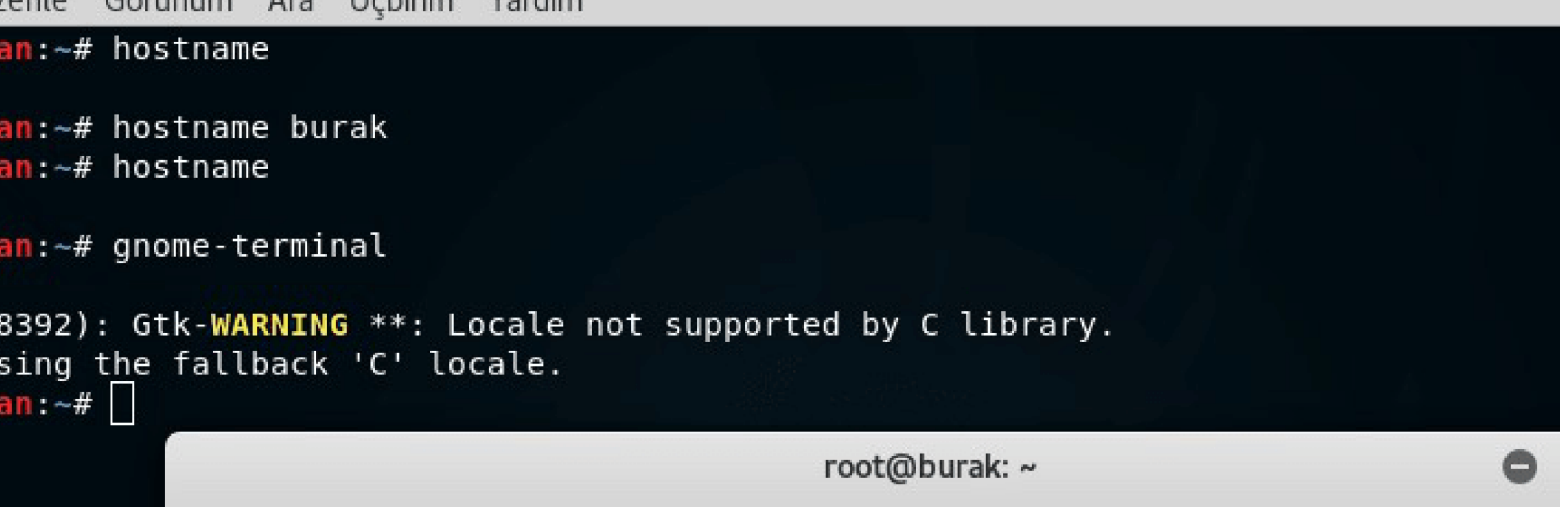
```
└─# uname --help
Kullanım: uname [SEÇENEK]...
Belirli sistem bilgilerini yaz. SEÇENEK olmadan -s ile aynı.

-a, --all                sıralamaya uyarak tüm bilgileri yaz,
                        eğer bilinmiyorsa -p ve -i atlama hariç:
-s, --kernel-name        çekirdek adını yaz
-n, --nodename            ana ağ bilgisayarının adını yaz
-r, --kernel-release     çekirdek sürümünü yaz
-v, --kernel-version     print the kernel version
-m, --machine            print the machine hardware name
-p, --processor           print the processor type (non-portable)
-i, --hardware-platform  print the hardware platform (non-portable)
-o, --operating-system    print the operating system
--help                  bu yardımı gösterir ve çıkar
--version               sürüm bilgisini gösterir ve çıkar

GNU coreutils online help:
Report any translation bugs to
Full documentation
or available locally via: info '(coreutils) uname invocation'
```

hostname

Bilgisayarımızın adını, diğer bir deyişle bilgisayarımızın ağ üzerindeki adını verir. Eğer isterseniz bu adı değiştirebilirsiniz. Örneğin; komut satırına **hostname burak** yazarsanız **hostname** yani **bilgisayarınızın adı burak** olarak değişmiş olacaktır.



```
root@taylan: ~
Dosya  Düzenle  Görünüm  Ara  Uçbirim  Yardım
root@taylan:~# hostname
taylan
root@taylan:~# hostname burak
root@taylan:~# hostname
burak
root@taylan:~# gnome-terminal

(process:8392): Gtk-WARNING **: Locale not supported by C library.
Using the fallback 'C' locale.
root@taylan:~#
```

```
root@burak: ~
Dosya  Düzenle  Görünüm  Ara  Uçbirim  Yardım
root@burak:~#
```

Ancak bu ayar kalıcı değildir. Oturumu kapatıp tekrar açtığınızda değiştirmiş olduğunuz adın tekrar eski haline döndüğünü görebilirsiniz. Kalıcı olmasını ayarlardan kolaylıkla sağlayabiliriz. Sürümden sürüme değişiklik gösterse de sırasıyla; ****Ayarlar > Sistem > Ayrıntılar > Genel aygıt adı: **** kutusuna istediğiniz ismi yazabilirsiniz. Eğer dediğim yoldan ulaşamadıysanız ayarları biraz kurcalayarak bulabilirsiniz.



Ayrıca tamamen komut satırından da **kalıcı bir hostname değişikliği** yapmak mümkün. Bunun için **/etc/hostname** konumunda yer alan **hostname** dosyasını açmamız gerek. Dosyamızı açmak için konsola **nano -w /etc/hostname** komutunu veriyoruz. Komutun ardından açılan dosyada ismimizin yerine istediğimiz ismi yazıyoruz. Ve **Ctrl + X** tuş kombinasyonu ile dosyamızdaki değişiklikleri kaydederek çıkıyoruz.

```
GNU nano 2.9.1 /etc/hostname Modified
taylan
^G Get Help  ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit      ^R Read File  ^\ Replace    ^U Uncut Text ^T To Spell   ^ Go To Line
```

Değişikliğin geçerli olması için oturumun kapatılıp tekrar açılması gerek daha sonra **hostname** yani **bilgisayarınızın ismi** değişmiş olacaktır.

lsb_release

Bu komut kullandığımız dağıtım hakkında farklı bilgiler sunan parametrelere sahiptir. Detaylı bilgi için **man** kılavuz sayfasını kontrol edebilirsiniz. Ben örnek olması açısından **lsb_release -a** komutunun çıktısını veriyorum.

```
└─$ lsb_release -a
No LSB modules are available.
Distributor ID: Kali
Description:    Kali GNU/Linux Rolling
Release:        2020.4
Codename:       kali-rolling
```

whoami

Sinemayı takip eden arkadaşların aklına ilk ne geldi biliyorum :)

Bu komut kimlik sorgu işlevi görüyor. Komut satırımıza komutumuzu vererek sonucuna bakalım.

```
└─$ whoami
root
```

Komut çıktısı bize mevcut kullanıcının **root** kimliği ile çalıştığını göstermiş oldu.

Yine **whoami** komutuna benzer ancak çok ufak farkların olduğu birkaç komutu daha peş peşe komut satırına girerek çıkan sonuçlar üzerinden anlatıma devam edelim.

```

root@taylan:~# who
root      tty3      2018-02-16 01:48
burak     :1         2018-02-16 01:58 (:1)
root      :2         2018-02-16 14:53 (:2)
root@taylan:~# w
 01:43:54 up 5:25, 9 users, load average: 0,06, 0,04, 0,01
USER      TTY      FROM            LOGIN@   IDLE   JCPU   PCPU WHAT
root      tty3      -               Cum01    23:54m 0.03s  0.02s -bash
burak     :1        :1             Cum01    ?xdm? 44.83s 0.01s /usr/li
root      :2        :2             14:53   ?xdm? 44.83s 0.01s /usr/li
root@taylan:~#

```

Sıra sıra bakalım:

whoami: mevcut kullanıcının hangi kimlikle çalıştığını gösteriyor.

who: sistemde hangi kullanıcının çalıştığını gösteriyor.

w: hangi kullanıcı hangi uygulamayı çalıştırıyor bunun bilgisini gösteriyor.

Bunun dışında bu komutlar da parametre alabiliyorlar. Hangi parametreleri aldıklarını nasıl öğrenebileceğinizi zaten biliyorsunuz.

uptime

uptime komutu **sistemimizin ne kadar zamandır açık olduğu bilgisini** verir.

```

└─$ uptime
13:31:32 up 15 min, 1 user, load average: 0,00, 0,04, 0,08

```

date

İsminden de anlaşılacağı gibi bu komut bize **sistemin o anki tarih ve saat bilgisini** veriyor.

```

└─$ date
Sa1 02 Şub 2021 13:32:58 EST

```

cal

Komutumuz Türkçe olarak takvim anlamına gelen İngilizce "**Calendar**" ifadesinin kısaltmasıdır. Komutumuz tek başına kullanıldığında bize konsol ekranı üzerinden, bulunduğumuz **tarihin takvim bilgisini** verir.

```

└─$ cal
      Şubat 2021
Pa Pz Sa Çr Pr Cu Ct
    1  2  3  4  5  6
  7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28

```

Ancak daha önceki veya sonraki tarihlerde yer alan takvim bilgilerine komutumuzu **cal ay yıl** şeklinde kullanarak ulaşabiliyoruz.

Örnek üzerinden açıklayalım.

Eğer 2002 yılının Ocak ayına gitmek istiyorsam komutumu **cal 1 2002** şeklinde kullanırım.


```
└─$ cal 1 2002
      Ocak 2002
Pa  Pz  Sa  Çr  Pr  Cu  Ct
      1   2   3   4   5
 6   7   8   9  10  11  12
13  14  15  16  17  18  19
20  21  22  23  24  25  26
27  28  29  30  31
```

Eğer ileri bir tarihe yani 3003 yılının Haziran ayına gitmek istersem de komutumu **cal 6 3003** şeklinde kullanırım.

```
└─$ cal 6 3003
      Haziran 3003
Pa  Pz  Sa  Çr  Pr  Cu  Ct
      1   2   3   4
 5   6   7   8   9  10  11
12  13  14  15  16  17  18
19  20  21  22  23  24  25
26  27  28  29  30
```

Gördüğünüz gibi ileri geri herhangi bir tarih hakkında takvim bilgisi almak istesek **cal** komutunu kullanabiliyoruz.

which

Herhangi bir komutun tam yol bilgisini öğrenmek için kullanılır.

```
└─$ which date
/usr/bin/date
```

whereis

Bu komutunda özel parametreleri vardır man sayfasından kontrol edin lütfen. Bunun dışında parametresiz hali **apropos** komutuna benzer şekilde bir çıktı verir. Ancak **apropos** komutundan farklı olarak **ilgili komutun man sayfası konumunun tam dizin adresini belirtir**.

```
└─$ whereis date
date: /usr/bin/date /usr/share/man/man1/date.1.gz
```

Ayrıca diğer parametrelerini de kesinlikle man sayfasından inceleyin.

locate

Bu komut ile aradığımız bir dosyanın nerede olduğunu öğrenebiliriz. Komutun çıktısı bize dosyanın bulunduğu dizin veya dizinleri veriyor. Örneğin daha önce **PATH** yoluna eklediğim **komut.sh** dosyasını **locate** komutu ile aratıyorum. Sonuç aşağıdaki şekilde:

```
└─$ locate komu.sh
/root/Desktop/yeni_dizin/yeni/en_yeni/komut.sh
```

Bu komutun ne kadar kıymetli olduğunu işiniz düştükçe ve kullandıkça anlayacaksınız. Genelde bir dosyayı arama çubuğu yardımı ile aramak sonuç vermez ve bu arama işlemi oldukça hantal çalışır. Örneğin dosya konumunu bilmediniz ancak sistem var olan bir dosyaya ulaşmak istiyorsunuz diyelim.. İşte tam bu noktada **locate** komutu

çıkageliyor ve bize saniyeler içerisinde sonucu veriyor. Komutu denemek amacıyla Kali Linux içerisinde yer alan bir araç olan **armitage** aracını aratıyorum. Bunun için komut satırına **locate armitage** yazmalıyım.

```
└─$ locate armitage
/usr/share/icons/Flat-Remix-Blue-Dark/apps/scalable/armitage.svg
/usr/share/icons/Flat-Remix-Blue-Dark/apps/scalable/kali-armitage.svg
/usr/share/icons/hicolor/16x16/apps/kali-armitage.png
/usr/share/icons/hicolor/22x22/apps/kali-armitage.png
/usr/share/icons/hicolor/24x24/apps/kali-armitage.png
/usr/share/icons/hicolor/256x256/apps/kali-armitage.png
/usr/share/icons/hicolor/32x32/apps/kali-armitage.png
/usr/share/icons/hicolor/48x48/apps/kali-armitage.png
/usr/share/icons/hicolor/scalable/apps/kali-armitage.svg
/usr/share/kali-menu/applications/kali-armitage.desktop
```

Çıktıda da görüldüğü gibi aradığımız ifadeyi içeren tüm dosyaların nerede olduğunu bir çırpıda buluverdik. Sanırım bu komut sayesinde yavaş yavaş da olsa komut satırının gücünü fark etmeye başlıyoruz :)

dmidecode

Eğer daha önce de Linux işletim sistemini deneyimleyip sorunlar ile karşılaştıysanız, forumlardan destek istediğinizde; muhtemelen insanlar sizden bu komutun çıktıları istemiş olabilirler. **Bu komutun işlevi sistemin donanım ve BIOS bilgilerini göstermektir.** Bu komut sayesinde sistem donanımı ve işleyişi hakkında çok fazla bilgiye ulaşabiliriz. Ayrıca bu komutumuz da parametre alarak çalışmaktadır.

Bu komutun sunduğu bilgiler çok fazla olduğu için, bilgilerin daha net anlaşılması adına; bu bilgilere belirli bir düzene sahip olan **DMI(Desktop Management Interface)** ile, yani **Masaüstü Yönetim Arayüzü** tabloları ile ulaşıyoruz. Tabloda yer alan sıralamayı ve hangi bilgileri alabileceğinizi direk olarak görmek isterseniz, komut satırına **dmidecode** yazın ve sonuçların kategorize şekilde karşınıza geldiğine kendiniz tanık olun.

Bahsi geçen tablo sıralı şekilde ve Numara-Türkçe karşılığı olacak şekilde aşağıdaki gibidir.

Numara	Açıklama
0	Bios
1	Sistem
2	Baz kurulu
3	Şasi
4	İşlemci
5	Bellek denetleyicisi
6	Bellek modülü
7	Önbellek
8	Port bağlantısı
9	Sistem yuvaları

Numara	Açıklama
10	On Board Cihazları
11	OEM Dizeleri
12	Sistem Yapılandırma Seçenekleri
13	BIOS Dili
14	Grup Dernekler
15	Sistem Event Log
16	Fiziksel Bellek Array
17	Bellek Cihazı
18	32-bit bellek hatası
19	Bellek Dizisi Haritalı Adres
20	Bellek Cihazı Haritalı Adres
21	Dahili İşaret Aygıtı
22	Taşınabilir Pil
23	Sistem Sıfırlama
24	Dananım Güvenlik
25	Sistem Güç Denetimleri
26	Gerilim Probu
27	Soğutma Cihazı
28	Sıcaklık Probu
29	Elektrik Akımı Probu
30	Uzaktan Erişim
31	Boot Bütünlüğü Hizmetleri
32	Sistem Önyükleme
33	64-bit Bellek Hatası
34	Yönetim Cihazı
35	Yönetimi Cihaz Bileşeni

Numara	Açıklama
36	Yönetimi Cihaz Eşik Verileri
37	Bellek Kanal
38	IPMI Cihazı
39	Güç Kaynağı

Başta belirttiğim gibi **dmidecode** kendi içinde parametre alan bir komut olduğu için istediğimiz spesifik bilgiye doğrudan da ulaşmamız mümkün. Detaylı bilgi için man kılavuz sayfasına bakın lütfen. Ancak küçük bir örnek vermem gerekirse bios hakkında bilgi edinmek istediğimizde konsola **dmidecode -t bios** komutunu girmemiz yeterli.

```
└─$ dmidecode -t bios
# dmidecode 3.3
Getting SMBIOS data from sysfs.
SMBIOS 2.5 present.

Handle 0x0000, DMI type 0, 20 bytes
BIOS Information
    Vendor: innotek GmbH
    Version: VirtualBox
    Release Date: 12/01/2006
    Address: 0xE0000
    Runtime Size: 128 kB
    ROM Size: 128 kB
    Characteristics:
        ISA is supported
        PCI is supported
        Boot from CD is supported
        Selectable boot is supported
        8042 keyboard services are supported (int 9h)
        CGA/mono video services are supported (int 10h)
        ACPI is supported
```

Burada ayrıca **dmidecode -t bios** komutu yerine, bios'un sıralamada bulunan numarasını yani **dmidecode -t 0** yazarak da aynı işlemi gerçekleştirebilirdik.

```
└─$ dmidecode -t 0
# dmidecode 3.3
Getting SMBIOS data from sysfs.
SMBIOS 2.5 present.

Handle 0x0000, DMI type 0, 20 bytes
BIOS Information
    Vendor: innotek GmbH
    Version: VirtualBox
    Release Date: 12/01/2006
    Address: 0xE0000
    Runtime Size: 128 kB
    ROM Size: 128 kB
    Characteristics:
        ISA is supported
        PCI is supported
        Boot from CD is supported
        Selectable boot is supported
        8042 keyboard services are supported (int 9h)
        CGA/mono video services are supported (int 10h)
        ACPI is supported
```

fdisk-l

Başlıkta da yer aldığı gibi bu bölümde **fdisk** komutunun yalnızca **l** parametresinin işlevini göreceğiz. Bu komutu burada vermemin sebebi sistem hakkında bilgi alırken diskin de sistem dahilinde olmasıdır. Zaten ileride tekrar ele alacağımızdan şimdilik bu kadarı da yeterli olacaktır. Bu komutu diskler üzerinde işlem gerçekleştirirken kullanıyoruz. Eğer komut satırımıza **fdisk -l** şeklinde komut verirse karşımıza sistemimizdeki disk bölümleri sıralı ve düzenli şekilde listelenecektir.

```
└─$ fdisk -l
Disk /dev/sda: 80 GiB, 85899345920 bytes, 167772160 sectors
Disk model: VBOX HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xe7875fa7

Device      Boot      Start          End      Sectors   Size Id Type
/dev/sda1   *          2048 165771263 165769216    79G  83 Linux
/dev/sda2                165773310 167770111    1996802   975M   5 Extended
/dev/sda5                165773312 167770111    1996800   975M  82 Linux swap / Solaris
```

df

Bu komut ile disk kullanımı hakkında ayrıntılı bilgiye ulaşabiliyoruz.

```
└─$ df
Dosyasistemi    1K-blok    Dolu      Boş  Kull% Bağlanılan yer
udev            985152      0    985152    0% /dev
tmpfs           203764     900    202864    1% /run
/dev/sda1       81058256 9767292 67130352   13% /
tmpfs           1018804      0   1018804    0% /dev/shm
tmpfs            5120        0     5120    0% /run/lock
tmpfs            4096        0     4096    0% /sys/fs/cgroup
tmpfs           203760     52    203708    1% /run/user/1000
```

Bu komutumuz da parametre alabiliyor, yani dileyen arkadaşlar konsolda **man df** şeklinde komut girerek gerekli bilgilere ulaşabilirler.

du

Bir dizinin, içerdiği tüm dosyalar ile birlikte diskte kapladığı toplam alanı verir. Ayrıca boyutla ilgili düzgün bir çıktı almak istersek **h** parametresini de kullanabiliriz.

```
└─$ du /root/Desktop
8      /root/Desktop/yeni_dizin/yeni/en_yeni
12     /root/Desktop/yeni_dizin/yeni
16     /root/Desktop/yeni_dizin
20     /root/Desktop
└─$ du -h /root/Desktop
8,0K   /root/Desktop/yeni_dizin/yeni/en_yeni
12K    /root/Desktop/yeni_dizin/yeni
16K    /root/Desktop/yeni_dizin
20K    /root/Desktop
```

free

Bu komut ile kullanılan bellek miktarını **KB** cinsinden öğrenebiliriz. Ancak çıktımızın **MB** cinsinden olmasını istersek **-m** parametresini **free** komutumuza ekleyerek kullanmamız yeterli olacaktır.

```
└─$ free
              total        used        free      shared  buff/cache   available
Mem:        2037612      463412      1139700         18000         434500      1411828
Swap:        998396           0         998396

└─$ free -m
              total        used        free      shared  buff/cache   available
Mem:           1989          452          1112           17           424          1378
Swap:           974           0           974
```

modinfo

Bu komut sayesinde Linux Kernel(çekirdek) modüllerinin bilgisi alınabilir. Bu modülleri ekran bastırmak isterseniz komut satırına **lsmod** yazarak modülleri listeleyebilirsiniz.

```
└─$ lsmod
Module                Size  Used by
fuse                  143360  3
rfkill                28672  2
vboxsf                40960  0
intel_rapl_msr        20480  0
intel_rapl_common     32768  1 intel_rapl_msr
intel_pmc_core_pltdrv  16384  0
intel_pmc_core        45056  0
rapl                  16384  0
..
.
```

Örneğin modüller içinden **battery** modülü hakkında bilgi almak istersek komut satırına **modinfo battery** şeklinde yazmamız yeterli olacaktır.

```
└─$ modinfo battery
filename:      /lib/modules/5.9.0-kali1-amd64/kernel/drivers/acpi/battery.ko
license:      GPL
description:   ACPI Battery Driver
author:       Alexey Starikovskiy
author:       Paul Diefenbaugh
alias:        acpi*:PNP0C0A:*
depends:
retpoline:    Y
intree:       Y
name:         battery
vermagic:     5.9.0-kali1-amd64 SMP mod_unload modversions
parm:         cache_time:cache time in milliseconds (uint)
```

stat

Bu komut dosyalar veya dizinler hakkındaki bilgileri almamıza olanak sağlar. Kullanımı **stat dizin** şeklindedir. Örneğin **root** dizinine bakalım. Bunun için komut satırımıza **stat /root** komutunu veriyoruz.

```
└─$ vmstat
File: /root
  Size: 4096          Blocks: 8          IO Block: 4096   dizin
Device: 801h/2049d   Inode: 4194305   Links: 4
Access: (0700/drwx-----)  Uid: (  0/   root)   Gid: (  0/   root)
Access: 2020-11-04 15:24:12.000000000 -0500
Modify: 2021-02-02 13:16:43.915999646 -0500
Change: 2021-02-02 13:16:43.915999646 -0500
Birth:  2020-11-17 07:32:57.373541909 -0500
```

Gördüğünüz gibi **root** dizini hakkında bilgiler basılmış oldu.

vmstat

Bu komut sayesinde sistemimizin o anlık genel durumunu görebiliriz. Ancak komutu verdikten sonra sistem durumu listelenip sonlanacaktır. Eğer sistemin durumunu bir süre izlemek istersek **vmstat gecikme_hızı yenilenme_sayısı** şeklinde komut vermeliyiz. Ben **her 2 saniyede bir yenilenmesini ve bunu 3 defa** yapmasını

istiyorum. Bunun için komut satırına **vmstat 2 3** komutunu veriyorum.

```
└─$ vmstat
procs -----memory----- ---swap-- -----io---- -system-- -----cpu-----
 r  b   swpd   free  buff  cache   si   so    bi    bo    in   cs  us  sy  id  wa  st
 1   0       0 1138816 40848 394112    0    0   63     4  309  109  0  0 99   1   0
└─$ vmstat 2 3
procs -----memory----- ---swap-- -----io---- -system-- -----cpu-----
 r  b   swpd   free  buff  cache   si   so    bi    bo    in   cs  us  sy  id  wa  st
 0   0       0 1138816 40856 394152    0    0   63     4  309  109  0  0 99   1   0
 0   0       0 1138816 40856 394152    0    0    0     0  641  157  0  0 100   0   0
 0   0       0 1138816 40856 394152    0    0    0     0  576  183  0  0 99   0   0
```

history

Bu kısma gelene kadar konsola bir çok komut yazdık. Peki o yazılan komutların daha sonra kullanılmak üzere saklandığını biliyor muydunuz ?

Evet girilen her komut **.bash_history** dosyasında tutuluyor. Biz bu dosyanın içeriğini yani daha önceki yazdığımız kodları görmek istersek komut satırına **history** yazmalıyız. Çıktı çok uzun olacağından çıktının tamamını aşağıda vermedim.

```
└─$ history
 1  dmidecode -t bios
 2  dmidecode -t 0
 3  fdisk -l
 4  df
 5  free
 6  free -m
 7  lsmod
 8  modinfo battery
 9  stat /root
10  vmstat
11  vmstat 2 3
12  history
```

Hazır liste uzun demişken eğer bu listenin limitini öğrenmek istersek komut satırına **echo \$HISTSIZE** yazarak bu değeri tutan ortam değişkeninin, ne kadar komutu **history** dosyasında tutulabileceğini görebiliriz.

```
└─$ echo $HISTSIZE
1000
```

Bu çıktı bize komutların tutulduğu dosyada en son **1000 komuta kadar** kayıt yapıldığını belirtiyor. Eğer tutulan komutların 1000 'den daha fazla olmasını isterseniz **.bashrc** dosyasında **HISTSIZE=1000** yazan değeri istediğiniz doğrultuda düzenleyebilirsiniz.

Komutun kullanımına geçmeden önce bu komutu sadece basit ve kısa komutların tekrar kullanılması olarak değerlendirmeyin. Örneğin sürekli kullandığınız çok uzun ve karmaşık bir komut düşünün. Bu komutun her defasında yazılması eziyet, bir yere kopyalanıp oradan tekrar tekrar kopyala-yapıştır şeklinde kullanılması ise hantallık olacaktır.

Komutun kullanımına gelecek olursak örnek üzerinden açıklayalım. Benim kayıt dosyamın bir kısmı aşağıdaki şekilde, ben burada yer alan **600. komutu** yani **vmstat** komutunu kullanmak istiyorum. Bunun için komut satırına **!600** yazmam yeterli olacaktır. Komutun kullanımı **!komut_numarası** şeklinde. Örnek için çıktıyı inceleyebilirsiniz.

```

592 fdisk -l
593 df
594 du
595 free
596 modinfo
597 lsmod
598 modinfo battery
599 stat /root
600 vmstat
601 vmstat 2 3
602 history
603 echo $HISTSIZE
604 vmstat
605 history
root@taylan:~# !600
vmstat
procs -----memory----- --swap-- -----io----- -system-- -----cpu-----
r b swpd free buff cache si so bi bo in cs us sy id wa st
1 0 0 2026384 143704 621300 0 0 26 5 45 104 1 0 99 0 0
root@taylan:~# vmstat
procs -----memory----- --swap-- -----io----- -system-- -----cpu-----
r b swpd free buff cache si so bi bo in cs us sy id wa st
0 0 0 2026412 143704 621288 0 0 26 5 45 104 1 0 99 0 0
root@taylan:~#

```

Daha önceden kullandığımız bir komutu aynı şekilde tekrar kullanmak için ise komut satırına **!komut** şeklinde komut girmeliyiz. Örnek vermek gerekirse daha önceki kısımlarda **stat** komutu ile **root** dizini hakkında bilgi edinmiştik. Ve bu bilgiye ulaşmak için **stat /root** komutunu kullanmıştık. Ben şimdi tekrar aynı komuta ulaşmak için komut satırına **!stat** yazıyorum ve çıktısı tıpkı **stat /root** yazdığımda olduğu gibi oluyor. Yani bu sayede komutun geri kalanını uzun uzun yazmak gibi bir dert kalmıyor.

```

└─$ !stat
stat /root
File: /root
Size: 4096          Blocks: 8          IO Block: 4096   dizin
Device: 801h/2049d Inode: 4194305    Links: 4
Access: (0700/drwx-----)  Uid: ( 0/   root)  Gid: ( 0/   root)
Access: 2020-11-04 15:24:12.000000000 -0500
Modify: 2021-02-02 13:16:43.915999646 -0500
Change: 2021-02-02 13:16:43.915999646 -0500
Birth: 2020-11-17 07:32:57.373541909 -0500

```

Eğer son kullandığımız komutu tekrar kullanmak istersek komut satırına **!!** yazmamız yeterli olacaktır.

```

└─$ !!
stat /root
File: /root
Size: 4096          Blocks: 8          IO Block: 4096   dizin
Device: 801h/2049d Inode: 4194305    Links: 4
Access: (0700/drwx-----)  Uid: ( 0/   root)  Gid: ( 0/   root)
Access: 2020-11-04 15:24:12.000000000 -0500
Modify: 2021-02-02 13:16:43.915999646 -0500
Change: 2021-02-02 13:16:43.915999646 -0500
Birth: 2020-11-17 07:32:57.373541909 -0500

```


Bunlar dışında da çok farklı kullanım şekilleri de mevcut. Bu bilgilere **man** komutu yardımı ile nasıl ulaşacağınızı zaten biliyorsunuz.

Ayrıca son yazılan komutlara ulaşmak için klavyede yer alan yön tuşlarından ↑ (yukarı) ve ↓ (aşağı) tuşlarını kullanarak önceki ve sonraki komutlarınıza ulaşabilirsiniz. Genellikle yön tuşlarını kullanmak bize anlık işlemlerimizde hız katmaktadır. Sizler de bu pratiklikleri mutlaka yeri geldikçe sıklıkla kullanacaksınız.



Alıştırmalar Hakkında

Yalnızca okumak yetmez, öğrendiğiniz bilgilerin kalıcı olabilmesi için bolca alıştırmaya yapmalısınız. Doküman içerisindeki bilgileri pekiştirmek için aşağıdaki alıştırmalar ile başlayabilirsiniz. Elbette burada yer alan alıştırmalar faaliyetleri dışında, konuyu öğrendiğinizi hissedene kadar kendiniz de bolca pratik yapmayı da ihmal etmeyin lütfen. Aksi halde öğrendiğiniz bilgiler kısa sürede unutulup gidecektir.

Kullanmakta olduğunuz **sistem çekirdeği hakkında** bilgi alın.

Bilgisayarınızın **ağ üzerindeki adını(hostname)** değiştirin.

Kullanmakta olduğunuz **dağıtım hakkında genel bilgi** almak için ilgili komutu kullanın.

Mevcut oturumda yani çalışmakta olduğunuz oturumdaki **kullanıcı kimliğini** konsol üzerinden ilgili komut yardımıyla sorgulayın.

"Sisteminizin **ne kadar süredir açık olduğunu** sorgulayın.

Konsoldan sistemin o anki **tarih ve saat bilgisini** sorgulayın.

Konsoldan **takvim bilgisini** sorgulayın.

Konsol üzerinden **2020 yılının mart ayı takvimini** görüntüleyin.

echo komutunun çalışmasını sağlayan dosyaların hangi dizinde tutulduğunu konsol üzerinden sorgulayın.

echo komutunun man kılavuz sayfalarında yer alan bilgilerinin hangi dosyada tutulduğunu **tam dizin adresi** şeklinde öğrenin.

"armitage" ifadesinin geçtiği dosyaları listeleyin.

Konsol üzerinden kullanılan **bellek(ram) miktarını MB**(megabayt) cinsinden öğrenin.

/root dizini hakkında genel bilgiler almak için ilgili komutu kullanın.

İlgili komut yardımıyla sisteminizin o anlık genel durumunu sorgulayın.

Son kullanmış olduğunuz komutu tekrar kullanmak için öğrendiğiniz şekilde pratik yöntemi uygulayın.

Konsola girmiş olduğunuz önceki komutları listeleyin.(komut geçmişinizi ilgili komut ile listeleyin).
Listelenen çıktılarından **10. satırdaki** komutu tekrar kullanmak üzere öğrendiğiniz şekilde **pratik yöntemi uygulayın**.

Aşağı yukarı yön tuşlarıyla önceki ve sonraki komutlar arasında geçiş yapın.

Geri Bildirimde Bulunun

Sizlere daha verimli bir kaynak sunabilmemiz için, uygulamada veya dokümantasyonlarda yer alan tüm hata ve eksiklerimizi bize bildirebilirsiniz.

Geri Bildirimde Bulunun