

***nix' lerde Etkin ve Pratik
Arama Yöntemleri**

```
/******  
* Bâkır EMRE *  
* emre [at] enderunix [dot] org *  
* EnderUNIX Yazılım Geliştirme Takımı *  
* http://www.enderunix.org *  
* * *  
* Sürüm : 1.0 *  
* Tarih : 20.02.2007 *  
* Etiketler : arama,find,locate,which,whereis,freebsd,linux *  
* Seviye : Başlangıç - Orta Seviye *  
* Makalenin en yeni versiyonu : http://www.enderunix.org/docs/etkinpratikarama.pdf *  
* adresinden elde edilebilir. *  
*****/
```

İçindekiler

Giriş.....	3
1. Locate	3
2. rlocate	5
3. find	7
4. which.....	11
5. whereis.....	11
Kaynaklar.....	13

Giriş

Unix sistemlerde dosya aramak için kullanabileceğimiz bir kaç yöntem vardır bunları kullanışlılık , cabuk sonuç vermesi doğru sonuç vermesi (dosyaların güncellenmemiş konumlarını göstermesi) bakımından değişiklik gösterir. Aşağıda kullanılan bir kaç program incelenmiş en son bölümde ise zaman karşılaştırmaları gösterilmiştir.

1.Locate

Locate yaptığı iş kısaca, aranacak olan dosyayı veya dizini daha önceden kaydettiği veritabanı üzerinden aramasından ibarettir.

```
/var/lib/mlocate/mlocate.db
```

```
/var/lib/slocate/slocate.db
```

Freebsd için locate kullanımında kullanılan komut ve dosyaların yerleri aşağıda listelenmiştir.

```
/var/db/locate.database      :   locate veritabanının varsayılan konumu  
/usr/libexec/locate.updatedb :   locate veritabanı güncelleyen script  
/etc/periodic/weekly/310.locate :   veritabanını yeniden oluşturan script
```

freebsd öntanımlı olarak locate veritabanını güncellemez

Freebsd locate veritabanını güncellemek için

```
/usr/libexec/locate.updatedb
```

linuxta güncellemek için

```
updatedb
```

komutunu çalıştırmalısınız yada yine freebsd' de

```
/etc/periodic/weekly/310.locate
```

altında yer alır periodic scripti vardır ve periodic komutunun çalıştırılmasıyla haftada bir olarak veritabanını günceller ki bu süre zarfında birçok dosya değişmekte silinmekte ve yeni oluşmaktadır dolayısıyla bu veritabanı eski kalacaktır siz bunu günlük olacak şekilde yapmak isterseniz

```
/etc/periodic/daily/
```

altında bir tane periodic scripti oluşturursanız en azından günlük veritabanınız olur

Bunun için

```
/etc/periodic.conf
```

dosyasına son eklediğiniz günlük locate scriptini de eklemeniz gerekir

```
# 305.locate
daily_locate_enable="YES"
```

diğer seçeneğimiz bahsettiğimiz gibi

```
/usr/libexec/locate.updatedb
```

komutu fakat bu komutu root olarak çalıştırırsanız bütün dosyalar indexlenecektir. locate Bunun olmasını engellemek için farklı veritabanları kullanmanıza izin vermektedir.

```
locate -d
```

parametresi farklı veritabanlarını oluşturmak için kullanılır.

Örnek kullanım olarak şu şekilde bir script işimiz görecektir.

Öncelikle locate için conf dosyası oluşturalım bu dosya linux için

```
/etc/updatedb.conf
```

içerisinde iken, freebsd' de

```
/usr/libexec/locate.updatedb
```

yani güncellemek için kullandığımız komutun kendisidir

```
FCODES="/root/locate/db/root_locate.veritabani"
```

Diye veritabanının yerini belirtelim daha sonra hangi klasörleri taraması gerektiğini atayalım

```
SEARCHPATH="/"
```

Biz burada / altındaki herşeyi indexlemesini istiyoruz

Hangi klasörleri taramaması gerektiğini belirtelim

```
PRUNEPATHS="/tmp /usr/tmp /var/tmp /var/db/portsnap /dev"
```

conf dosyasına bunları yazdıktan sonra,

Ortam değişkeni olarak bu ayar dosyasını shell ekleyelim

```
env -i LOCATE_CONFIG=/root/locate/conf/locate.rc \
/usr/libexec/locate.updatedb
```

root kullanıcıasına ait locate veritabanını /root dizini altında oluşturalım

Varsayılan locate veritabanı ise /var/db/locate.database şeklindedir.

```
locate -d /root/locate.veritabani
```

root kullanıcı için her defasında root' a has veritabanını yerini belirtmemek için alias tanımlayabiliriz

```
alias locate locate -d /root/locate.veritabani
```

diye...

böylece root kullanıcısı bütün dosyaları görebilecekken sıradan kullanıcıların bütün dosyaların görmesi engellenecektir. Aşağıda ise sıradan kullanıcılar için oluşturulan locate veritabanının güncellenmesi ve haklarının belirtilmesi gösterilmektedir

```
#!/bin/sh
```

```
touch /var/db/locate.database
```

```
chown nobody /var/db/locate.database
```

```
chmod 644 /var/db/locate.database
```

```
cd /
```

```
echo "/usr/libexec/locate.updatedb | su - -fm "nobody"
```

locate in kullanışlı olmasına karşın yukarıda görüldüğü gibi güncellenme sorunu vardır

2. rlocate

locate in güncellenme sorununu çözmesine karşın sadecelinux 2.6 çekirdekler üzerinde çalışır. Daemon yada kernel modulu olarak çalışabilmektedir. Kaynak kodundan derledikten sonra

Kernel modulu olarak

```
modprobe rlocate
```

eğer

```
FATAL: Error inserting rlocate (/lib/modules/2.6.17-11-generic/misc/rlocate.ko): Invalid argument
```

şeklinde bir hata alıyorsanız SELinux özelliğini kapatmanız gerekiyor kullandığınız linux dağıtımına göre selinux.conf dosyasına disable yazmanız yada

/boot/grub/menu.lst

dosyasına

```
kernel /boot/vmlinuz-2.6.17-11-generic root=/dev/sda5 ro
single selinux=0
```

gibi satır sonuna

```
selinux=0
```

yazmanız gerekmektedir

çekirdek modulu olarak yukarıdaki şekilde yükledik. Eğer programın daemon olarak başlatılmasını istiyorsak

```
/etc/init.d/rlocate start
```

komutu verilerek yüklenir.

```
/etc/cron.daily/rlocate
```

komutu verilerekten güncel kalması sağlanır.

Aşağıda ise parametreleriyle beraber örnek bir iki kullanım gösterilmektedir.

Verilen klasör için indexleme yapmak için

```
rlocate -U <dir>
```

verilen klasörleri indexlememesi için -e parametresi ve klasörlerin ismi yazılır.

```
rlocate -e "/proc,/dev/,/tmp" -u
```

verilen dosya sistemleri üzerindeki dosyaların indexlenmemesi için -f parametresi kullanılır.

```
rlocate -f "NFS,iso9660" -u
```

3. find

En basit kullanımından başlarsak

```
find aranacakdizin -parametre nearanacaksao
```

```
find . -name aradığımızdosya
```

bulduğumuz dizinde **aradığımızdosya** adındaki dosyaları listeler

```
find / -name neredebudosya
```

/ dizininden itibaren bütün dosyaları arar **neredebudosya** adındaki dosyaları listeler. Aynı şekilde birden fazla yer gösterilip te arama yapılabilir

```
find /usr/src /home/emre/kaynak_kodlar/ -name arp.c  
  
/home/emre/kaynak_kodlar/src/contrib/ipfilter/ipsend/arp.c  
  
/usr/src/lib/libstand/arp.c
```

Diğer parametlerine gelince

- **-iname** n : name ile aynı fakat küçük – büyük harfe duyarlı
- **-atime** n : n gün önce erişilmiş dosyaları bulmak için kullanılır.
- **-amin** n : n dakika içerisinde erişilmiş dosyaları bulmak için kullanılır.
- **-mtime** n : n gün içerisinde içeriği değişmiş dosyalar.
- **-mmin** n : n dakika içerisinde içeriği değişmiş dosyalar.
- **-ctime** n : n gün içerisinde dosya özellikleri değişmiş dosyalar.
- **-cmin** n : n dakika içerisinde dosya özellikleri değişmiş dosyalar

burada n parametresinden evvel kullanılacak + - kullanılabilir ve komutun anlamını değiştirir.

-atime -n : n gün önce ile günümüz arasındaki erişim yapılmış dosyalar

-atime +n : n günden daha eski erişilmiş dosyalar

Şu şekilde örnek verecek olursak

```
find /etc -mmin -10
```

/etc klasörü altındaki dosya içeriği on dakika içerisinde değişmiş dosyaların listesi verir.

ayrıca dosya boyutuna göre arama yapmak istersek size parametresini kullanmamız gerekmektedir

```
find / size +500k -name dosyaadi
```

boyutu 500k dan fazla olan **dosyaadi** adındaki dosyaları bulmak için kullanabiliriz.

Aynı şekilde

```
find /root size -300k
```

root altındaki boyutu 300k dan küçük olan dosyaları listeler.

```
find . -size +100M -fprintf /home/emre/buyuk_dosyalar.txt '%-10s
%p\n'
```

yukarıdaki komut ise bulunulan dizindeki 100 Mb dan büyük dosyaları bulup buyuk_dosyalar.txt adlı dosyaya satır satır yazmaktadır.

size dan sonra gelen sayı ile beraber “c” kullanırsak karakter sayısına göre arama yapmış olacağız

```
find . -size +50000c
```

içerisinde 50000 karakter olan dosyaları listeler

Eğer find i normal kullanıcı olarak çalıştırırsanız listelemeye hakkınız olmayan klasörlerde hata verecektir.

```
$ find / -name "*arıyorumbulamıyoru*"
```

```
find: /lost+found: Permission denied
find: /var/log/setup/tmp: Permission denied
find: /var/log/setup/apache: Permission denied
find: /var/man/cat1: Permission denied
```

Bu hataları almamak (daha doğrusu gizlemek için:) için /dev/null a yönlendirebilirsiniz.

```
$ find / -name "*arıyorumbulami*" 2> /dev/null
```

```
/usr/local/share/arıyorumbulami.conf
/home/kullaniciadim/arıyorumbulamıyorum.mp3
```

...

Şimdide söyle bişeyler deneyelim diyelimki mp3lerimiz bir klasör içerisinde ve bunun bir üst dizininde mp3 dosyası varmı .Bunu şu şekilde ararsak

```
$ find /home/kullaniciadim -name "*.mp3"
```

```
/home/kullaniciadim/mepeuclerim/1.mp3
/home/kullaniciadim/mepeuclerim/2.mp3
/home/kullaniciadim/mepeuclerim/3.mp3
/home/kullaniciadim/mepeuclerim/4.mp3
/home/kullaniciadim/mepeuclerim/5.mp3
/home/kullaniciadim/mepeuclerim/6.mp3
/home/kullaniciadim/mepeuclerim/numaraatladi.mp3
/home/kullaniciadim/mepeuclerim/7.mp3
/home/kullaniciadim/mepeuclerim/yemedi.mp3
```



```
/home/kullaniciadim/mepeuclerim/8.mp3
/home/kullaniciadim/mepeuclerim/9.mp3
. . .
```

bu komutu su sekilde calistirirsak

```
$ find /home/kullaniciadim -path "mepeuclerim/" -prune -name
"* .mp3"
/home/kullaniciadim/birtaneburadakalmis.mp3
/home/kullaniciadim/baskabidizin/birtanedeburadakalmis.mp3
```

Tam olarak istediğimiş şeyi yapmış olacağız. Parametrelere gelince **-path** parametresiyle hangi klasör olduğu belirtildi ve **-prune** kullanarak bu dizini es geçtik ve **mepeuclerim** klasörü dışında ama **/home/kullaniciadim** altındaki bütün dosyaları bu şekilde bulabiliriz

Find in diğer parametrelerine gelince

Dosya tipine göre arama yapmak için **-type** kullanılabilir.

Aşağıda belirtilen seçenekler ile beraber kullanılır.

b	block (buffered) special
c	character (unbuffered) special
d	directory
p	named pipe (FIFO)
f	regular file
l	symbolic link
s	socket
D	door (Solaris)

Örnek kullanımı

socketleri listelemek istersek

```
find /tmp -type s

/tmp/ksocket-root/kdeinit__0
/tmp/ksocket-root/kdeinit-:0
/tmp/ksocket-root/klauncherC133F4.slave-socket
/tmp/mcop-root/bilgisayarim_bizimev-02e8-45b8b09c
/tmp/.ICE-unix/dcop725-1169731733
/tmp/.ICE-unix/749
/tmp/.X11-unix/X0
/tmp/.s.PGSQL.5432
```

Sadece klasor aramak istersek

```

find /etc/ -type d
    /etc/
    /etc/defaults
    /etc/X11
    /etc/X11/wmconfig
.....
find: /etc/isdn: Permission denied
    /etc/mail
    /etc/mtree
    /etc/ntp
find: /etc/ntp: Permission denied
    /etc/pam.d
    /etc/periodic
    /etc/periodic/daily
.....
    /etc/periodic/weekly
    /etc/ppp
    /etc/rc.d
    /etc/skel
    /etc/ssh
...
    /etc/.java/deployment
    /etc/security

```

eğer boş dosyaları ya da klasörleri bulmak istersek

```

find . -empty
    /etc/ntp
    /etc/skel
    /etc/dumpdates
    /etc/opekeys

```

parametresiyle kullanabiliriz.

Bir dosya değilde dosya içerisinde geçen bir kelimeyi arıyorsanız

```

find / -type f -exec grep "text" {} /dev/null \;

```

dosya izinlerine görede arama yapmamız mümkün bunu için **-perm** parametresini kullanacağız.

```

find . -perm 777 -exec chmod 755 {} \;

```

yukarıdaki örnekte izni 777 olan dosyaları listeleyip bunları yeni bi parametre olan **-exec** ile bu dosyaların modlarını 755 olarak değiştiriyoruz

```

find / -perm -0002 -type f -print

```

yukarıdaki komut ise yazılabilir bütün dosyaları listelemektedir

4. which

Çevre değişkenlerinde belirtilen (PATH) dosyaları aramak için kullanılır. Dosyanın tam yolunu belirtir. a parametresiyle aranan dosya için bütün yolları gösterir.

```
which -a mono  
  
/usr/bin/mono  
/usr/bin/X11/mono
```

5. whereis

Aranacak olan program, man sayfaları ya da kaynak klasörlerin yerini veren komuttur.

ikili dosyaların dosya yolu için -b parametresi

```
whereis -b mono  
  
/usr/local/bin/mono
```

kaynak kodlar için -s parametresi

```
whereis -s echo  
  
/usr/src/bin/echo
```

man sayfaları için -m parametresi

```
whereis -m perl  
  
/usr/local/man/man1/perl.1.gz
```

kullanılabilir. u parametresi kullanılmayan

aşağıdaki komut ise /usr/bin altında manual sayfaları olmayan programları listeler

```
whereis -m -u /usr/X11R6/bin/*  
  
aclocal-1:  
acpi_fakekey:  
activation-client:  
afmdiff:  
akodeplay:
```

amsn :
amsn-remote :

...

gibi

yukarıda gördüğümüz komutların kullanıldığı yerlere göre arama yaparsak sonuçları daha etkin ve hızlı şekilde arayabiliriz.

örneğin arayacağımız bir komut (program) ise **which** i kullanmamız daha mantıklıdır

time locate locate

time whereis locate

time which locate

locate	whereis	which
real 0m0.057s	real 0m0.031s	real 0m0.002s
user 0m0.039s	user 0m0.002s	user 0m0.000s
sys 0m0.016s	sys 0m0.009s	sys 0m0.002s

find için path belirtmek gerektiği için yukarıdaki listeye eklemedim

aramak istediğimiz şey sıradan bir dosya ise locate in find dan hızlı olmasının karşın findin parametreleri ile daha etkin bir sonuç elde edebiliriz.

time find /root -name *.pdf

time locate *.pdf

find	locate
real 0m1.238s	real 0m0.042s
user 0m0.027s	user 0m0.041s
sys 0m0.063s	sys 0m0.001s

Kaynaklar

1- <http://rlocate.sourceforge.net/>

2- man find

3- man whereis

4- man locate