HOWTO zum Kommando "find"

(C) 2006-2019 T.Birnthaler/H.Gottschalk <howtos(at)ostc.de>
OSTC Open Source Training and Consulting GmbH
http://www.ostc.de

$Id: find-HOWTO.txt,v 1.42 2020/02/05 00:41:35 tsbirn Exp $

Dieses Dokument beschreibt die Kommandos "find" und "locate".

INHALTSVERZEICHNIS

---

1) Beschreibung

Das Ã¤uÃM-^_erst mÃ¤chtige Kommando "find" durchsucht VerzeichnisbÃ¤ume rekursiv nach Dateien mit bestimmten Eigenschaften. Suchkriterien kÃ¶nnen neben dem Dateinamen alle (im Inode vorhandenen) Dateiattribute (Metadaten) wie Dateityp, Besitzer, Besitzergruppe, Zugriffsrechte, DateilÃ¤nge, Zeitstempel, ... sein. Diese Suchkriterien sind beliebig Ã¼ber AND, OR, NOT und Klammerung kombinierbar.

"find" listet entweder alle zu den Bedingungen CONDITION passenden Dateinamen auf (Standardverhalten) oder fÃ¼hrt beliebige Kommandos ACTION auf ihnen durch. Dazu wird der Verzeichnisbaum ab den angegebenen Verzeichnissen PATH rekursiv durchlaufen (das kann einige Zeit dauern). Die allgemeine Aufrufsyntax lautet:

```
  find [PATH...] [CONDITION...] [ACTION...]
```

HINWEIS: Bei fehlendem Suchpfad PATH beginnt die Suche im aktuellen Verz. ".". Bedingungen CONDITION sind per Default UND-verknÃ¼pft (d.h. mÃ¼ssen ALLE erfÃ¼llt sein), fehlt CONDITION gilt sie als erfÃ¼llt. Die Standard-Aktion ACTION ist "-print" (d.h. alle ab Verz. PATH zu den Bedingungen CONDITION passenden Dateinamen werden ausgegeben). D.h. ein Aufruf der Form

```
  find
```

fÃ¼hrt zum Auflisten ALLER Verz.- und Dateinamen ab dem aktuellen Verzeichnis.

1.1) Die wichtigsten Bedingungen CONDITION

Bedingung	Bedeutung
-mtime [+-]DAY	Inhalt-Ã M-^Dnderung vor +mehr/-weniger/genau Tagen [modify]
-name "PATTERN"	Dateiname passt zu Pattern (* ? [...] ... schÃ¼tzen!)
-perm [-+/]RIGHT	Oktale Rechte (-000=mind, +000 bzw. /000=any, 000=exakt) 1=x=Execute, 2=w=Write, 4=r=Read [permission]
-size [+-]NUM... ...[GMkwcb]	DateigrÃ¶Ã M-^_e (+mehr/-weniger/genau) in 512-Byte BlÃ¶cken (bzw. G=Giga M=Mega k=Kilo w=2-byte c=byte b=512-Byte)
-type TYPE	Dateityp (f=file d=directory l=symboliclink D=door c=characterdev. b=blockdev. p=named pipe s=socket)
-group GNAME/GID	Besitzergruppe (Name oder GID)
-user UNAME/UID	Besitzer (Name oder UID)

Bei Zahlenwerten (DAY, NUM) bedeutet die Zahl:
   +N = GrÃ¶Ã M-^_er/mehr als angegebener Wert (gleich nicht enthalten!)
   -N = Kleiner/weniger als angegebener Wert (gleich nicht enthalten!)
    N = GENAU der angegebene Wert

Bei "-perm" (RIGHT) bedeutet PrÃ¤fix:

```
   /NNNN = Mind. EINES der angegebenen Rechte NNNN vorhanden (modern)
   +NNNN = Mind. EINES der angegebenen Rechte NNNN vorhanden (veraltet)
   -NNNN = Mind. ALLE der angegebenen Rechte NNNN vorhanden
    NNNN = Genau ALLE der angegebenen Rechte NNNN vorhanden


1.2) Weitere wichtige Bedingungen CONDITION
-------------------------------------------

  +-----------------+-------------------------------------------------------+
  | Bedingung       | Bedeutung                                             |
  +-----------------+-------------------------------------------------------+
  | -atime [+-]DAY  | Lesender Zugr. vor +mehr/-weniger/genau Tagen [access]|
  | -ctime [+-]DAY  | Status-Ã¤nderung (Inode) vor +mehr/... Tagen [change] |
  +-----------------+-------------------------------------------------------+
  | -iname "PATTERN"| Dateinamen-Muster (* ? [...] ... vor Shell schÃ¼tzen!),|
  |                 |   GroÃ-^_/Kleinschreibung egal [ignorecase]            |
  | -path "PATTERN" | Pfadnamen-Muster (* ? [...] ... vor Shell schÃ¼tzen!)  |
  | -ipath "PATTERN"|   Analog, aber GroÃ-^_/Kleinschreibung egal [ignorecase]|
  +-----------------+-------------------------------------------------------+
  | -inum NUM       | Inode-Nummer NUM                                      |
  | -links [+-]NUM  | Anzahl Hardlinks +mehr/-weniger/genau NUM (Anzahl Namen)|
  +-----------------+-------------------------------------------------------+
  | -nogroup        | KEINER Gruppe in "/etc/groups" zugeordnet (GID ohne Name)|
  | -nouser         | KEINEM Benutzer in "/etc/passwd" zugeord. (UID ohne Name)|
  +-----------------+-------------------------------------------------------+


1.3) Bezugszeitpunkt von Datumvergleichen
-----------------------------------------

Beim Datumvergleich per -mtime/-atime/-ctime DAY (bzw. -mmin/-amin/-cmin MIN)
ist die Basis fÃ¼r die Tages/MinutenzÃ¤hlung der STARTZEITPUNKT des Kommandos
"find" (d.h. eine DYNAMISCHE Definition!). Die Bedeutung der "Tage" DAY bzw.
der Minuten MIN ist daher:

  0 =  0:00:00h-23:59:59h VOR Startzeitpunkt (bzw. vor 00:00 bei -daystart)
  1 = 24:00:00h-47:59:59h VOR Startzeitpunkt (bzw. vor 00:00 bei -daystart)
  2 = 48:00:00h-71:59:59h VOR Startzeitpunkt (bzw. vor 00:00 bei -daystart)
  3 = 72:00:00h-95:59:59h VOR Startzeitpunkt (bzw. vor 00:00 bei -daystart)
  ...
                                                        |
                                          Startzeitpunkt | 00:00 akt. Tag
                                                        |   (-daystart)
                                                        |
                        VERGANGENHEIT                   |   ZUKUNFT
                                                        |
  ...|<= 6 =>|<= 5 =>|<= 4 =>|<= 3 =>|<= 2 =>|<= 1 =>|<= 0 =>|<= -1 =>|...
  ...|168-144|144-120|120-96h| 96-72h| 72-48h| 48-24h| 24-00h|
  <========== +3 ==========>|       |<========= -3 ======================>
  ...|        |<============= +0 und -6 =============>|       |
  ...|<=================== -7 (akt. Woche + Zukunft!) ======================>
                                                        |

HINWEIS: Option "-daystart" legt die Basis des Zeitvergleichs auf den Beginn
00:00 des aktuellen Tages fest (statt auf den Startzeitpunkt von "find").


1.4) MÃ¶gliche Aktionen ACTION
-----------------------------

  +-----------------+-------------------------------------------------------+
  | Aktion          | Beschreibung                                          |
  +-----------------+-------------------------------------------------------+
  | -print          | Ausgabe gefundener Dateinamen (Standard!)             |
  | -ls             | Ausgabe analog "ls -dils" (directory inode long size) |
  | -exec CMD {} \; | Kommando CMD auf gefundenen Dateien ausfÃ¼hren ({} = gef.|
  |                 |   Dateiname, \; = Kommando-Ende, quotieren wg. Shell) |
  | -ok CMD {} \;   | Analog "-exec", verlangt vorher BestÃ¤tigung mit "y"(es)|
  | -delete         | Datei lÃ¶schen (Vorsicht!)                             |
  +-----------------+-------------------------------------------------------+


1.5) VerknÃ¼pfung von Bedingungen CONDITION
------------------------------------------

  +---------+-----------------------------------------------------------+
  | Verknpf | Beschreibung                                              |
  +---------+-----------------------------------------------------------+
  | \(...\) | Klammerung (quotieren wg. Shell!)                         |
  | \!      | Negation (quotieren wg. Shell!)                           |
  | -a      | UND-VerknÃ¼pfung (Standard falls keine VerknÃ¼pfung angegeben!) |
```

-o	ODER-VerknÃ¼pfung

HINWEIS: Die VerknÃ¼pfungen sind nach fallender Vorrang angegeben, durch Klammerung lÃ¤sst sich dieser Vorrang durchbrechen.

1.6) NÃ¼tzliche "find"-Beispiele

Beispiel	Beschreibung (Std: Ergebnis ausgeben)
find -name "*.c"	C-Dateien "*.c" ab akt. Verz. "."
find . -name "*.c"	(analog)
find .. -name "*.c"	C-Dateien "*.c" ab Eltern-Verz. "."
find \! -name "*.c"	Alle auÃM-^_er C-Dateien "*.c" ab "."
find ~ -name "*.c"	C-Dateien "*.c" ab Heimat-Verz. "~"
find / -name "*.c"	C-Dateien "*.c" ab Root-Verz. "/"
find / -iname "*.c"	analog + GROSS/kleinschreibung ignor.
find / -regex ".*\.c"	analog per RegulÃ¤rem Ausdruck
find / -iregex ".*\.c"	analog + GROSS/kleinschreibung ignor.
find /usr -type d -name "*man*"	Verz. in "/usr" mit "man" im Namen
find /usr -type d -regex ".*man.*"	analog per RegulÃ¤rem Ausdruck
find /usr /lib /etc -name "*.o"	Objekt-Dateien "*.o" in 3 angeg. Verz.
find / -mtime 0	Heute geÃ¤nderte Dateien
find / -mtime 1	Gestern geÃ¤nderte Dateien
find / -mtime -7	Letzte Woche (Tage 0-6) geÃ¤nd. Dateien
find / -mtime +6 -mtime -14	Vorletzte Woche (Tage 7-13) geÃ¤nd. Dat.
find / \(-mtime 7 -o -mtime 8 \ -o -mtime 9 -o -mtime 10 \ -o -mtime 11 -o -mtime 12 \ -o -mtime 13 \)	(analog: Tage einzeln aufzÃ¤hlen und per ODER verknÃ¼pfen)
find / -atime +365	Letzter Zugriff vor mehr als 1 Jahr
find / -atime +365 -daystart	(analog, aber Startzeitpunkt 00:00)
find / -newer FILE	Neuer als Datei FILE
find / -user tsbirn	Dateien des Anwenders "tsbirn"
find / -user root -perm -002	GehÃ¶ren root + fÃ¼r andere schreibbar
find / \(-nouser -o -nogroup \)	GehÃ¶ren keinem User oder keiner Gruppe
find / -uid +99 -uid -201	GehÃ¶ren User mit 100 <= UID <= 200
find / -gid +0 -gid -100	GehÃ¶ren Gruppe mit 1 <= GID <= 99
find / \(-uid 1000 -o -gid 1000 \) -print0 | xargs -0 chown 1005:1005	UID/GID 1000 --> 1005 konvertieren
find / -type f \(-perm -100 -o -perm -010 -o -perm -001 \)	AusfÃ¼hrbare Dateien (Programme)
find / -type f -perm -4000	Dateien mit SetUID-Recht (mindestens)
find / -type f -perm -2000	Dateien mit SetGID-Recht (mindestens)
find / -type d -perm -2000	Verz. mit SetGID-Recht (mindestens)
find / -type f -perm -1000	Dateien mit Sticky-Recht (mindestens)
find / -type f -inum 123	Dateien zu Inode 123
find / -type f \! -type l -links +1	Dateien mit Anz. Hardlinks > 1
find / -type f -exec grep -il "made" {} \;	Dateien die Text "made" enthalten
find / -type f | xargs grep -il "made"	(analog)
grep -ilR "made" /	(analog per grep "rekursiv" aufrufen)
find / -type f -size 0 -ok rm {} \;	Leere Dateien lÃ¶schen (mit Abfrage)
find / -type f -empty -ok rm {} \;	(analog)
find / -type d -exec rmdir -p {} \;	Leere Verz. lÃ¶schen (kein Inhalt)
find / -type d -empty -exec ...	Leere Verz. lÃ¶schen (besser!)
find -L / -type l -exec rm {} \;	Broken Symlinks lÃ¶schen (Ziel ex. nicht)
find / -xtype l -delete	(analog)

TIPP: Alle Dateien im eigenen Heimatverz. auflisten, die in der vorletzten Woche (Tag 7-13) verÃ¤ndert wurden und den Text "muster" enthalten (GROSS/kleinschreibung egal):

```
find ~ -type f -mtime +6 -mtime -14 \               # f=file, m=modify
       -exec grep -li "muster" {} \; 2> /dev/null   # -l=list -i=ignorecase
```

1.7) GNU-"find" kennt weitere Bedingungen CONDITION

Bedingung	Bedeutung

-P	Symbol. Links NIE verfolgen (Standard!)
-L / -follow	Symbol. Links IMMER verfolgen [links, dereference]
-H	Symbol. Links NICHT verfolgen AUSSER Kmdo-Argumente
-d / -depth	Erst. Verz.inhalt., dann Verz. selbst bearbeiten (Standard: Erst Verz. selbst, dann Verz.inhalt)
-mount / -xdev	Nur Verz. auf GLEICHEM Dateisystem betrachten
-prune	NICHT in Verz. absteigen
-noleaf	NICHT optimieren dass "." + ".." vorhanden bei -name...
-maxdepth LEV	Max. LEV Verz. + Dateien absteigen (0=nur Kmdo-Argumente)
-mindepth LEV	Verz. + Dateien bis LEV ignorieren (1=Kmdo-Argument ign.)
-fstype TYPE	Filesystemtyp (mÃ¶gl. Werte per -printf %F, z.B. "ext3")
-mmin [+-]MIN	ÃM-^Dnderung vor +mehr/-weniger/genau MIN Minuten [modify]
-amin [+-]MIN	Lesender Zugriff vor +mehr/... MIN Minuten [access]
-cmin [+-]MIN	Status-ÃM-^Dnderung (Inode) vor +mehr/... MIN Minuten [change]
-newer FILE	Inhalt-ÃM-^Dnderung VOR ÃM-^Dnderung an FILE [modify]
-anewer FILE	Lesender Zugriff VOR lesendem Zugriff auf FILE [access]
-cnewer FILE	Status-ÃM-^Dnderung (Inode) VOR Status-ÃM-^Dnd. an FILE [change]
-newerXY FILE	X: a=access, B=birth, c=inode change, m=modification time Y: t=FILE als Zeitangabe im "date -d" Format interpretiert
-used [+-]DAY	Zugriff +mehr/-weniger/genau DAY Tage seit ÃM-^Dnderung
-uid [+-]UID	Analog "-user" aber per Nummer (Bereich von/bis mÃ¶glich)
-gid [+-]GID	Analog "-group" aber per Nummer (Bereich von/bis mÃ¶glich)
-empty	Leere Datei/Verz. (leeres Verz. sonst schwer zu prÃ¼fen)
-regex "PATTERN"	RegulÃ¤rer Ausdruck passt [regular expression]
-iregex "PATTERN"	Analog, aber GroÃM-^_/Kleinschr. egal [ignorecase]
-wholename "PAT"	Ganzer Pfadnamen gemÃ¤ÃM-^_ Shell-Muster "PATTERN" passt
-iwholename "PAT"	Analog, aber GroÃM-^_/Kleinschr. egal [ignorecase]
-samefile "FILE"	Gleiche Inode-Nummer wie FILE (also Hardlink auf FILE)
-daystart	Tagesbeginn 00:00 statt "find"-Startzeitpunkt als Basis!

HINWEIS: -regex ... muss VOLLSTÃM-^DNDIG auf Pfad passen (^ und $ sind wegzulassen!)
-E schaltet extended Regex (ERE) ein \(\) -> ( ) ? + {...}
Teillmatch daher so zu schreiben: -regex ".*REGEX.*"

1.8) GNU-"find" kennt weitere Aktionen ACTION

Bedingung	Bedeutung
-print0	Namen in gen. Liste durch "\0" (NUL-Bytes) trennen (Standard: durch "\n", fÃ¼r "xargs -0" sinnvoll)
-printf FMT	Benutzerdef. Ausgabe mit %-Formatelementen in FMT
-fls FILE	Analog "-ls", aber auf Datei FILE ausgeben
-fprint FILE	Analog "-print", aber auf Datei FILE ausgeben
-fprint0 FILE	Analog "-print0", aber auf Datei FILE ausgeben
-fprintf FILE FMT	Analog "-printf FMT", aber auf Datei FILE ausgeben

Bei "-printf" und "-fprintf" sind folgende Formatbezeichner in FMT angebbar:

Fmt	Bedeutung	
%a	Letzter Zugriffszeitpunkt	[access]
%b	GrÃ¶ÃM-^_e in 512-Byte BlÃ¶cken	[block]
%c	Letzte StatusÃ¤nderung	[change]
%d	Tiefe im Verz.baum (0=Kommando-Argument)	[depth]
%D	GerÃ¤tenummer	[Device]
%f	Reiner Dateiname (letzte Komponenten) ohne Pfadanteil	[filename]
%F	Dateisystemtyp	[File]
%g	Besitzergruppenname (oder GID)	[group]
%G	Besitzergruppen-ID (GID)	[GID]
%h	Pfadname ohne Dateiname (letzte Komponente)	[dirpath]
%H	Kommando-Argument das Fund der Datei ausgelÃ¶st hat	
%i	Inode-Nummer	[inode]
%k	GrÃ¶ÃM-^_e in 1K BlÃ¶cken	[kilo]

%l	Ziel eines Symb. Links (leer falls keiner)	[link]	
%m	Zugriffsrechte in Oktalnotation	[permission]	
%M	Zugriffsrechte in symbolischer Form	[Permission]	
%n	Anzahl Hardlinks auf Datei	[hardlink]	
%p	VollstÃ¤ndiger Dateiname	[path]	
%P	Dateinamen ohne Kommando-Argument das Fund der Datei auslÃ¶ste	[Path]	
%s	DateigrÃ¶Ã-^_e in Byte	[size]	
%S	Leerheit von Dateien (1.0=keine, <=1.0)	[Sparse]	
%t	Letzte Inhalt-Ã-^Dnderung	[change time]	
%u	Besitzername (oder UID)	[user]	
%U	Besitzer-ID (UID)	[UID]	
%y	Dateityp "fdlcbpxD" (analog "ls -l")	[type]	
%Y	Dateityp mit Verfolgung symb. Links (L=Loop, N=Nichtexistent)	[type]	
%%	Prozentzeichen "%"		
\n	Zeilenvorschub (analog \r \t \v \f \\)		

1.9) GNU-"find" kennt weitere VerknÃ¼pfungen

Verkn	Bedeutung
-and	Analog "-a" (Standard falls keine andere VerknÃ¼pfung angegeben!)
-or	Analog "-o"
-not	Analog "!" ("!" ist normalerweise zu schÃ¼tzen: "\!"
,	Erst linke Seite (Ergebnis verwerfen) dann rechte Seite (Ergebnis)

2) Performance-Problem von "find"

"find" startet das nach "-exec/-ok" angegebene Kommando fÃ¼r JEDE gefundene Datei neu. Dies ist bei sehr vielen gefundenen Dateien ineffektiv, da jedesmal ein neuer Prozess erzeugt wird.

```
find $HOME -type f -size +200 -exec gzip {} \;
```

Effektiver sind die folgenden Aufrufe (einfÃ¼gen des Ergebnisses von "find" auf der Kommandozeile per Kommandosubstitution `...` oder $(...) bzw. sammeln der MAXIMAL mÃ¶glichen Menge an Argumenten vor einem Aufruf von "gzip" per "xargs"):

```
gzip `find $HOME -type f -size +200 -print`          # 1) sh
gzip $(find $HOME -type f -size +200 -print)         # 2) bash, ksh
find $HOME -type f -size +200 -print | xargs gzip    # 3) xargs
```

Allerdings kann bei den Varianten 1)+2) ein "Ã-^\berlaufproblem" eintreten, wenn die einzufÃ¼gende Liste zu lang ist (abhÃ¤ngig von der Shell ab 1-2 Mio Zeichen oder mehreren 1000 Namen).

Alle obigen Varianten 1)-3) haben den Nachteil, dass bei bestimmten Sonderzeichen in Dateinamen (Whitespace, ...) die erzeugte Liste falsch interpretiert wird. Bei GNU-"find" und GNU-"xargs" gibt es dafÃ¼r ein LÃ¶sung, die als Trennzeichen das NUL-Byte "\0" verwendet (statt Zeilenvorschub "\n"), das PRINZIPIELL nicht in einem Dateinamen vorkommen kann (neben dem "/"):

```
find $HOME -type f -size +200 -print0 | xargs -0 gzip       # oder
find $HOME -type f -size +200 -print0 | xargs --null gzip   #
```

HINWEIS: Viele weitere Kommandos, die Listen von Dateinamen verarbeiten, kennen inzwischen diesen Schalter -0/--null (z.B. tar, cpio, grep, sed, ...).

3) "find"-Ersatz "locate"

"locate" ist eine schnellere, aber nicht so leistungsfÃ¤hige Variante von "find" (kann nur nach DATEINAMEN suchen, nicht aber nach sonstigen Dateiattributen). Sucht nicht im Dateibaum direkt, sondern in Indexdatei "/var/lib/locatedb".

```
locate "PATTERN"   # Datei gemÃ¤Ã-^_ Muster PATTERN suchen (* ? [...] ...)
```

Diese Indexdatei muss vorher mit "updatedb" erzeugt und von Zeit zu Zeit aktualisiert werden (da Ã-^Dnderungen am Dateisystem darin nicht vermerkt werden). Dies kann einige Zeit dauern und wird meist tÃ¤glich um Mitternacht Ã¼ber einen "crontab"-Job erledigt (wenn der Rechner zu diesem Zeitpunkt lÃ¤uft). "updatedb" kann aber auch per Hand aufgerufen werden (mit root-Rechten!):

```
sudo updatedb      # locate-Datenbank updaten
```