	Grundlagen in Linux find		Linux-UE5_1 1. Semester
Name:	Klasse:	Datum:	Blatt Nr.: 1/5 lfd. Nr.:

UE-5_1 Dateien suchen – find

Das Kommando `find` sucht den Verzeichnisbaum rekursiv nach Dateien ab, die den angegebenen Kriterien entsprechen. »Rekursiv« bedeutet, dass es auch alle im Startverzeichnis enthaltenen Unterverzeichnisse mit erfasst. Als Suchergebnis liefert `find` die Pfadnamen der gefundenen Dateien, die dann an andere Programme weitergeleitet werden können. Zur Verdeutlichung der Befehlsstruktur ein Beispiel:

```
$ find . -user test -print
./liste
```

Hier wird also das aktuelle Verzeichnis inklusive aller Unterverzeichnisse nach Dateien durchsucht, die dem Benutzer `test` gehören. Die Anweisung `-print` dient dazu, das Suchergebnis, im Beispiel nur eine einzige Datei, auf dem Bildschirm auszugeben. Zur Arbeitserleichterung wird daher die Angabe `-print` automatisch ergänzt, wenn Sie nicht ausdrücklich gesagt haben, wie mit den gefundenen Dateinamen zu verfahren ist.

Wie anhand des Beispiels zu sehen ist, benötigt `find` einige Angaben, um seine Aufgabe ordnungsgemäß erledigen zu können.

Startverzeichnis

Die Auswahl des Startverzeichnisses sollte mit Bedacht erfolgen. Wenn Sie das Wurzelverzeichnis wählen, wird die gesuchte Datei – sofern sie denn existiert – mit Sicherheit gefunden werden, der Suchvorgang kann jedoch viel Zeit in Anspruch nehmen. Natürlich dürfen Sie nur in denjenigen Verzeichnissen suchen, auf die Sie angemessene Zugriffsrechte haben.

Ausgabe absolut oder relativ

Die Angabe eines absoluten Pfadnamens für das Startverzeichnis liefert als Suchergebnis absolute Pfadnamen, ein relativ angegebenes Startverzeichnis ergibt entsprechend relative Resultate.


Verzeichnisliste

Statt eines einzelnen Startverzeichnisses können Sie auch eine Liste von Verzeichnisnamen angeben, die dann der Reihe nach durchsucht werden.

Auswahlkriterien

Mit diesen Angaben können Sie die Merkmale der zu findenden Dateien genauer festlegen. Tabelle „*Testkriterien von find*“ enthält eine Auswahl zulässiger Angaben, weitere stehen in der Dokumentation.

Wenn Sie mehrere Auswahlkriterien gleichzeitig angeben, werden diese automatisch Und-verknüpft, es müssen also alle erfüllt werden. Daneben versteht `find` noch andere logische Operationen (siehe Tabelle: „*Logische Operatoren für find*“).

	Grundlagen in Linux find		Linux-UE5_1 1. Semester
Name:	Klasse:	Datum:	Blatt Nr.: 2/5 lfd. Nr.:

Kompliziertere logische Verknüpfungen von Auswahlkriterien können (bzw. sollten) in runde Klammern eingeschlossen werden, um fehlerhafte Auswertungen zu vermeiden. Die Klammern müssen Sie natürlich vor der Shell verstecken:

```
$ find . \( -type d -o -name "A*" \) -print
./
./..
./bilder
./Abfall
$ _
```


Das Beispiel listet alle Dateien auf dem Bildschirm auf, die entweder Verzeichnisse repräsentieren oder deren Name mit einem »A« beginnt oder beides.

Aktion

Die Ausgabe der Suchergebnisse auf dem Bildschirm geschieht, wie eingangs erwähnt, mit der Kommandooption `-print`. Daneben existieren mit `-exec` (engl. *execute*, »ausführen«) und `-ok` noch zwei weitere Kommandos, die es ermöglichen, Folgekommandos mit den gefundenen Dateien auszuführen. Hierbei unterscheidet sich `-ok` nur dadurch von `-exec`, dass vor der Ausführung des Folgekommandos der Benutzer, um Zustimmung gebeten wird; `-exec` setzt diese stillschweigend voraus. Im Weiteren Text steht daher `-exec` stellvertretend für beide Varianten.

Testkriterien von find

Test	Beschreibung
-name	Sucht nach passenden Dateinamen. Hierbei sind alle Sonderzeichen nutzbar, die von der Shell zur Verfügung gestellt werden. Mit <code>-iname</code> werden dabei Groß- und Kleinbuchstaben gleich behandelt.
-type	Ermöglicht die Suche nach verschiedenen Dateitypen Dabei werden unterschieden: b blockorientierte Gerätedatei c zeichenorientierte Gerätedatei d Verzeichnis f normale Datei l Symbolisches Link p FIFO (named pipe) s Unix-Domain-Socket
-user	Sucht nach Dateien, die dem angegebenen Benutzer gehören. Dabei ist statt des Anwendernamens auch die Angabe der eindeutigen Benutzernummer, der UID, möglich.

	Grundlagen in Linux find		Linux-UE5_1 1. Semester
Name:	Klasse:	Datum:	Blatt Nr.: 3/5 lfd. Nr.:


-group	Sucht nach Dateien, die zu der angegebenen Gruppe gehören. Wie bei <code>-user</code> ist hier statt des Gruppennamens auch die Angabe der eindeutigen Gruppennummer, der GID, zulässig.
-size	Sucht nach Dateien bestimmter Größe. Zahlenwerte werden dabei als 512-Byte-Blöcke interpretiert, durch ein nachgestelltes c sind Größenangaben in Byte, durch k in Kibibyte erlaubt. Vorangestellte Plus- oder Minuszeichen entsprechen Unter- und Obergrenzen, womit ein Größenbereich abgedeckt werden kann. Das Kriterium <code>-size +10k</code> trifft z. B. für alle Dateien zu, die größer als 10 KiB sind.
-atime	(engl. access, »Zugriff«) sucht Dateien nach dem Zeitpunkt des letzten Zugriffs. Hier sowie für die beiden nächsten Auswahlkriterien wird der Zeitraum in Tagen angegeben; <code>... min</code> statt <code>... time</code> ermöglicht eine minutengenaue Suche.
-mtime	(engl. modification, »Veränderung«) wählt passende Dateien über den Zeitpunkt der letzten Veränderung aus.
-ctime	(engl. change, »Änderung«) sucht Dateien anhand der letzten Änderung der Inodes (durch Zugriff auf den Inhalt, Rechteänderung, Umbenennen etc.)
-perm	Findet nur Dateien, deren Zugriffsrechte genau mit den angegebenen übereinstimmen. Die Festlegung erfolgt mittels einer Oktalzahl, die beim Befehl <code>chmod</code> beschrieben wird. Möchte man nur nach einem bestimmten Recht suchen, muss der Oktalzahl ein Minuszeichen vorangestellt werden, z. B. berücksichtigt. <code>-perm -20</code> alle Dateien, die Gruppenschreibrechte besitzen, unabhängig von deren übrigen Zugriffsrechten.
-links	Sucht nach Dateien, deren Referenzzähler den angegebenen Zahlenwert hat.
-inum	Findet Verweise auf die Datei mit der angegebenen Inode-Nummer.

Logische Operatoren für find

Zeichen	Operator	Bedeutung
!	Nicht	die folgende Bedingung darf nicht zutreffen
-a	and (Und)	die Bedingungen links und rechts vom -a müssen zutreffen
-o	or (Oder)	von den Bedingungen links und rechts vom -o muss mindestens eine zutreffen

Zur Anwendung von `-exec` gibt es einige allgemeingültige Regeln:

- Das Kommando hinter `-exec` ist mit einem Strichpunkt (`» ; «`) abzuschließen. Da dieser in üblichen Shells eine Sonderbedeutung hat, muss er maskiert werden (etwa als `» \ ; «` oder mit Anführungszeichen), damit `find` ihn überhaupt zu sehen bekommt.
- Zwei geschweifte Klammern (`» { } «`) werden in dem Kommando durch den gefundenen Dateinamen ersetzt. Am besten setzen Sie die geschweiften

	Grundlagen in Linux find		Linux-UE5_1 1. Semester
Name:	Klasse:	Datum:	Blatt Nr.: 4/5 lfd. Nr.:

Klammern in Anführungszeichen, um Probleme mit Leerzeichen in Dateinamen zu vermeiden.

```
$ find . -user test -exec ls -l '{}' \;
-rw-r--r-- 1 test users 4711 Oct 4 11:11 datei.txt
$ _
```

Das obenstehende Beispiel sucht nach allen Dateien innerhalb des aktuellen Verzeichnisses, die dem Anwender test gehören und führt für diese das Kommando »ls -l« aus. Mehr Sinn ergibt da schon folgendes:

```
$ find . -atime +13 -exec rm -i '{}' \;
```

Hiermit werden alle Dateien im aktuellen Verzeichnis und darunter interaktiv gelöscht, auf die seit zwei Wochen oder länger nicht mehr zugegriffen wurde.


Mitunter – etwa im letzten Beispiel – ist es sehr ineffizient, für jeden einzelnen Dateinamen extra einen neuen Prozess mit dem `-exec`-Kommando zu starten. In diesem Fall hilft oft das Kommando `xargs`, das so viele Dateinamen wie möglich sammelt, bevor es damit tatsächlich ein Kommando ausführt:

```
$ find . -atime +13 | xargs -r rm -i
```

`xargs` liest seine Standardeingabe bis zu einem bestimmten (konfigurierbaren) Maximum von Zeichen oder Zeilen und verwendet das Gelesene als Argumente für das angegebene Kommando (hier `rm`). Als Trennzeichen für die Argumente gelten dabei Leerzeichen (die mit Anführungszeichen oder »\« maskiert werden können) oder Zeilentrenner. Das Kommando wird nur so oft wie nötig aufgerufen, um die Eingabe zu verbrauchen. – Die Option »-r« von `xargs` sorgt dafür, dass das Kommando `rm` nur ausgeführt wird, wenn `find` wirklich einen Dateinamen schickt – ansonsten würde es zumindest einmal gestartet.

Die `find/xargs`-Kombination kommt bei eigenartigen Dateinamen in Schwierigkeiten, etwa solchen, die Leerzeichen oder gar Zeilentrenner enthalten, welche dann als Trennzeichen fehlinterpretiert werden. Die todsichere Abhilfe dagegen besteht darin, die `find`-Option »-print0« zu benutzen, die wie `-print` die Namen der gefundenen Dateien ausgibt, diese aber nicht durch Zeilentrenner, sondern durch Nullbytes voneinander trennt. Da das Nullbyte in Dateinamen nicht auftauchen kann, ist keine Verwechslung mehr möglich. `xargs` muss mit der Option »-0« aufgerufen werden, um diese Form der Eingabe zu verstehen:

```
$ find . -atime +13 -print0 | xargs -0r rm -i
```

	Grundlagen in Linux find		Linux-UE5_1 1. Semester
Name:	Klasse:	Datum:	Blatt Nr.: 5/5 lfd. Nr.:

Übungen

1. Suchen Sie nach der „ssh“!
2. Suchen Sie nach Dateien, die größer sind als 50kb im /etc Verzeichnis!
3. Suchen Sie alle Dateien, die Ihrem User gehören und lassen Sie sich dazu die Rechte etc. anzeigen!
4. Finden Sie alle Dateien in Ihrem System, die größer als 1 MiB sind, und lassen Sie deren Namen ausgeben.
5. Wie können Sie find benutzen, um eine Datei zu löschen, die einen merkwürdigen Namen hat (etwa mit unsichtbaren Kontrollzeichen oder mit Umlauten, die von älteren Shells nicht verstanden werden)?
6. Wie würden Sie beim Abmelden dafür sorgen, dass etwaige Dateien in /tmp, die Ihnen gehören, automatisch gelöscht werden?