

Protokoll - myfind

Mit dem Programm myfind kann man eine einfache Dateisuchfunktion umsetzen. Es erzeugt, anders als die klassischen Varianten, für jeden angeforderten Dateinamen einen eigenen Kindprozess, der die Suche eigenverantwortlich ausführt. Die Resultate werden danach auf der Standardausgabe angezeigt

Aufruf und Parameter

Die Anwendung kann wie folgt gestartet werden:

myfind [-R] [-i] [Dateiname2] ... [DateinameN]

-R durchsucht rekursiv alle Unterverzeichnisse.

-i beachtet beim Vergleich von Namen nicht, ob ein Buchstabe groß oder klein geschrieben ist. Mindestens ein Startverzeichnis sowie ein Dateiname sind erforderlich.

Vorgehensweise

1. Argumentbearbeitung

Die Optionen werden mit getopt ausgewertet, und die restlichen Argumente werden als Suchpfad und Zielnamen übernommen.

2. Aufteilung der Prozesse

Das Hauptprogramm ruft für jeden Dateinamen fork() auf.

Im Kindprozess erfolgt die Suche nach genau diesem einen Namen. Der Elternprozess speichert die PIDs und wartet dann mit waitpid auf das Ende aller Kindprozesse.

3. Dateisuche

Es wird überprüft, ob der Suchpfad existiert und gültig ist.

Je nach Option wird entweder nur das Startverzeichnis oder der gesamte Unterbaum durchlaufen.

Berücksichtigt werden nur reguläre Dateien.

Ein Treffer tritt ein, wenn der Dateiname (gegebenenfalls ohne Berücksichtigung der Groß-/Kleinschreibung) mit dem gesuchten Namen übereinstimmt.

4. Ausgabeformat

Jeder gefundene Datensatz wird dargestellt als : : .

Es wird damit eindeutig ersichtlich, welcher Prozess den Fund gemeldet hat und wo sich die Datei befindet.

5. Synchronisierung

Da mehrere Kindprozesse gleichzeitig schreiben können, wird die Standardausgabe über flock jeweils kurz exklusiv gesperrt. Dadurch erscheinen die Trefferzeilen in beliebiger Reihenfolge, jedoch stets vollständig und unvermischt.

Fehlerbehandlung

Bei ungültigen Aufrufen erfolgt eine Ausgabe zur Usage.

Nicht vorhandene oder nicht zugängliche Pfade führen zu Fehlermeldungen.

Ausnahmen in den Kindprozessen werden aufgefangen, um sicherzustellen, dass sie das übrige Programm nicht beeinflussen.

Fazit

Die geforderte Funktionalität wird vom Programm erfüllt: parallele Dateisuche nach Namen, optionale Rekursion, Groß-/Kleinschreibung ignorierender Vergleich sowie atomare Ausgabezeilen. Die Aufteilung in Kinderprozesse ermöglicht eine unabhängige Bearbeitung der Suchanfragen. Zudem wird durch die Sperrung der Standardausgabe eine Konsistenz der Ausgabe gewährleistet.