

Aufgabe 6: myfind (12 Punkte)

27.11.2008

Implementieren Sie ein Programm **myfind**, das ähnlich dem UNIX-Kommando **find** arbeitet. Das Programm wird wie folgt aufgerufen:

```
myfind path... [-maxdepth n] [-name pattern] [-type {d,f}] [-links n]
```

Das Programm erhält einen oder mehrere Pfadnamen (Dateinamen oder Verzeichnisnamen) auf der Kommandozeile, optional gefolgt von Optionen und Suchausdrücken. Alle Pfade werden rekursiv in einer Tiefensuche durchlaufen. Hierbei werden alle Verzeichniseinträge ignoriert, bei denen es sich nicht um ein Verzeichnis oder eine Datei handelt, sowie die speziellen Einträge '.' und '..'. Pfadnamen von Einträgen, auf die **alle** angegebenen Suchausdrücke passen, sollen ausgegeben werden. Werden keine Suchausdrücke angegeben, so soll alles ausgegeben werden. Lesen Sie zunächst **find(1)** und experimentieren Sie mit UNIX-**find**, insbesondere mit den oben angegebenen Optionen und Suchausdrücken. Ihr Programm soll sich ähnlich verhalten, wobei nicht alle Optionen des "echten" **find** unterstützt werden.

a) Maximale Tiefe der Suche (-maxdepth *n*)

Die **maxdepth** Option erlaubt die Begrenzung der Suchtiefe auf ein frei wählbares Level $n \geq 0$ (0: Nur die auf der Kommandozeile übergebenen Pfade selbst werden untersucht (nicht deren evtl. Inhalt), 1: der Inhalt auf der Kommandozeile übergebener Verzeichnisse wird untersucht, aber nicht der ihrer Unterverzeichnisse, usw.). Ist diese Option nicht spezifiziert soll Ihr Programm bis an die Blätter der Verzeichnishierarchie absteigen. Sie können davon ausgehen, dass die max. Pfadtiefe die max. erlaubte Zahl von offenen Dateien eines Prozesses nicht übersteigt (andernfalls soll sich das Programm mit einer Fehlermeldung beenden).

b) Suchausdrücke

Ihr Programm soll eine Reihe ausgewählter Suchausdrücke unterstützen. Nur Pfadnamen, die alle angegebenen Bedingungen erfüllen, werden ausgegeben. Achtung: Die Suchausdrücke schränken nur die Ausgabe von find ein. Wenn also die Suchausdrücke nicht auf ein gefundenes Verzeichnis passen, die maximale Pfadtiefe aber noch nicht erreicht ist, so wird dieses Verzeichnis dennoch durchsucht.

- **name**: Mit diesem Ausdruck kann der Name eines Verzeichniseintrages mit einem Shell-Wildcard-Muster verglichen werden. Verwenden Sie für den Vergleich die Funktion **fnmatch(3)**.
- **type**: Hiermit kann die Suche auf Verzeichnisse (**d**) oder reguläre Dateien (**f**) eingegrenzt werden.
- **links**: Hiermit kann nach Einträgen mit einer bestimmten Zahl von Hardlinks gesucht werden.

Hinweise:

- Weitere Manpages: **opendir(3)**, **readdir(3)**, **closedir(3)**, **lstat(2)**, **atoi(3)**
- Zum Testen ihres Programms eignet sich z.B. das Verzeichnis */usr*.
- Zur Verwendung von **lstat(2)** müssen Sie die zusätzliche Compiler-Option **-D_XOPEN_SOURCE=500** verwenden.

Abgabe: bis spätestens Donnerstag, 11.12.2008, 14:00 Uhr