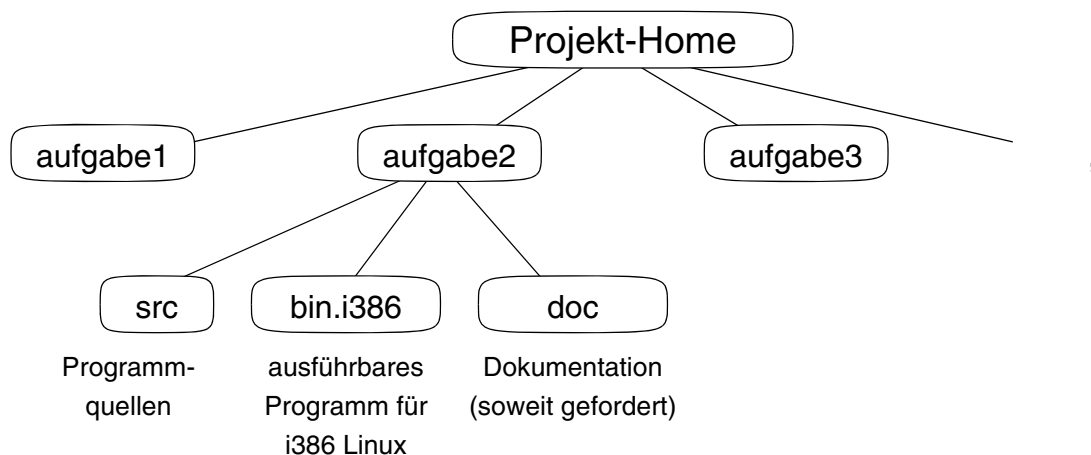


Allgemeine Hinweise zu den Übungen SOS I

Der Übungsbetrieb der Wiederholerübungen findet in einer Mischform aus Tafel- und Rechnerübung statt, in der je nach Bedarf vom Übungsbetreuer der Stoff erläutert wird oder die Teilnehmer Lösungen am Rechner erarbeiten. Bei Bedarf können die Teilnehmer dazu aufgefordert werden, ihre Lösung an der Tafel zu präsentieren und zu erläutern. Falls ein Teilnehmer seine Abgabe nicht erläutern kann, so wird die Aufgabe als nicht abgegeben bewertet. Zusätzlich werden durch einen Plagiattest die Abgaben der Teilnehmer, auch aus früheren Semestern, verglichen. Bei starker Übereinstimmung wird die Aufgabe ebenfalls als nicht abgegeben gewertet. Hierzu kann im Zweifelsfall ein persönliches Gespräch außerhalb der Übung zur Klärung geführt werden.

Der Directory-Baum für die Aufgaben ist folgendermaßen aufzubauen:



- Die Teilbäume mit den Übungsaufgaben dürfen nicht für andere Benutzer zugreifbar sein. Das **Projekt-Home** ist im Verzeichnis `/proj/i4sos/$USER` zu finden.
- Die Aufgaben sind bis spätestens zum Abgabetermin durch Aufruf des Programms
`/proj/i4sos/pub/abgabe aufgabeX` $X=1 \dots n$
abzugeben. Dieses Programm überprüft die Directory-Struktur und die Namen der Dateien, die laut der Aufgabenstellung vorhanden sein müssen, und erzeugt dann ein Archiv der abgegebenen Dateien. Bis zum Abgabetermin kann ein Programm beliebig oft abgegeben werden — es gilt der letzte (vor dem Abgabetermin vorgenommene) Aufruf des Abgabeprogramms.
- In den Wiederholerübungen finden keine Miniklausuren statt.

Aufgabe 1: myfind (12 Punkte)

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, 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.

b) Suchausdrücke

Ihr Programm soll eine Reihe ausgewählter Suchausdrücke unterstützen. Nur Pfadnamen, die alle angegebenen Bedingungen erfüllen, werden ausgegeben.

- **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.

c) Kommandozeilenparameter

Das Parsen von Kommandozeilenparametern ist eine häufig wiederkehrende und oft lästige Aufgabe; daher existieren für die meisten Programmiersprachen Bibliotheken, die diese Aufgabe vereinfachen. Verwenden Sie in Ihrem Programm die Funktion **getopt_long_only(3)**.

Hinweise:

- Achten Sie auf eine gute Struktur Ihres Programmes, da es später erweitert werden soll.
- Beachten Sie bitte auch die Informationen zur Abgabe und den Abgabezeitpunkt. Informationen dazu finden Sie unter http://www4.informatik.uni-erlangen.de/Lehre/SS08/V_SOS1/
- Weitere Manpages: **opendir(3)**, **readdir(3)**, **closedir(3)**, **lstat(2)**
- Zum Testen ihres Programms eignet sich z.B. das Verzeichnis */usr*.