SWO 3x Übung zu Softwareentwicklung mit klassischen Sprachen u. Bibliotheken 3 **WS 2014/15, Übung 04**

Abgabetermin: Sa in der KW 46

☐ **Gr. 1,** DI Franz Gruber-Leitner **Name** Roman Lumetsberger **Aufwand in h** 6

☒ **Gr. 2,** Dr. Erik Pitzer

Punkte _____ **Kurzzeichen Tutor / Übungsleiter** _____ / _____

1. Stdlib & Find (8 + 4 + 6 + 6 Punkte)

Die Standardbibliothek stellt sehr viel an Funktionalität bereits zur Verfügung. Um sie effizient zu verwenden, muss man die vorhandene Dokumentation aber sorgfältig studieren. Um das zu üben sollen Sie in diesem Beispiel eine einfache Version des UNIX Werkzeugs `find` implementieren, das einen Verzeichnisbaum rekursiv durchläuft und Dateien mit bestimmten Eigenschaften sucht, bzw. verarbeitet, wie Sie in der *manpage* von find nachlesen können.

(a) Implementieren Sie dazu im ersten Schritt mit Hilfe der Standardbibliothek eine Funktion, die, ausgehend von einem Startverzeichnis, rekursiv alle Verzeichnisse und Dateien durchläuft und für alle regulären Dateien mit einer beliebigen Funktion verarbeitet, die als Funktionszeiger übergeben wird, also z.B. folgende Schnittstelle erfüllt:

```
typedef void (*Visitor)(char *pathname, struct stat *stat);
void walkDir(char *dirname, Visitor visitor);
```

Die übergebene Funktion `visitor`, erhält also für jede Datei, den vollständigen Dateinamen, sowie die Dateiattribute (`man lstat`).

(b) Die jeweilige Visitor-Funktion soll per Kommandozeilenargument ausgewählt werden können. Es soll dabei berücksichtigt werden, dass manche dieser Visitor-Funktionen ein zusätzliches Argumente enthalten können. Es kann angenommen werden, dass alle möglichen Visitor-Funktionen bereits bekannt sind und somit hartcodiert werden können.

(c) Implementieren Sie eine erste Visitor-Funktion, die einfach alle Dateien und deren wichtigste Attribute ausgibt. Es soll mindestens folgendes ausgegeben werden:

- vollständiger Dateiname mit Pfad
- letztes Änderungsdatum
- Berechtigungen
- Größe

Eine mögliche Ausgabe könnte z.B. so aussehen:

```
user@ubuntu:~/swo3/homeworks/u4$ find .. -print
../u1/Makefile~           rwxrwxrwx      136 09/15/14 14:21
../u1/prime               rwxrwxrwx     7581 09/18/14 14:03
../u1/prime.c             rwxrwxrwx      951 09/18/14 14:03
../u1/prime.c~            rwxrwxrwx      949 09/15/14 14:09
../u1/triangle            rwxrwxrwx     7493 09/18/14 14:03
../u1/triangle.c          rwxrwxrwx      733 09/18/14 14:03
../u1/triangle.c~         rwxrwxrwx      732 09/15/14 14:21
                             .
                             .
                             .
../u4/find                rwxrwxrwx    21743 10/20/14 15:59
../u4/find.c              rwxrwxrwx     4635 10/20/14 15:59
../u4/find.c~             rwxrwxrwx     4620 10/20/14 15:59
../u4/find.o              rwxrwxrwx    17552 10/20/14 15:59
../u4/Makefile            rwxrwxrwx      108 10/20/14 13:10
../u4/Makefile~           rwxrwxrwx      109 10/20/14 13:10
```

(d) Implementieren Sie eine zweite Visitor-Funktion, die alle Dateien nach einer bestimmten Zeichenkette durchsucht und nur passende Zeilen ausgibt, z.B:

```
user@ubuntu:~/swo3/homeworks/u4$ find . –grep failed
find.c:115:21 "    printf("failed to open directory \"%s\"\n", dirname);"
find.c:123:25 "        printf("failed to stat \"%s\"\n", filename);"
find.c~:115:21 "    printf("failed to open directory \"%s\"\n", dirname);"
find.c~:123:25 "        printf("failed to stat \"%s\"\n", filename);"
```

Hinweis: Um Sie nicht beim Aufstöbern der Dokumentation verzweifeln zu lassen, hier noch eine Liste mit potentiell nützlichen Funktionen:

- `dirent(), readdir()` um Verzeichnisse zu traversieren
- `stat(), S_ISDIR()` um Datei- und Verzeichniseigenschaften abzufragen
- `localtime(), strftime()` um das Datum zu formatieren
- `fopen(), getline()` um Dateien zeilenweise zu lesen
- `strstr(), strlen(), strcpy(), strcat(), strcmp()` Funktionen von Zeichenketten

Außerdem hilfreich könnte das Studium den *manpage* über *man* selbst hilfreich sein (`$ man man`), sowie das Kommando `apropos` zum Finden von relevanten *manpages*.