SPiCAufgabe #7: `printdir`

(12 Punkte, Zweier-Gruppen)

Entwickeln Sie ein Programm `printdir`, das - ähnlich wie das UNIX-Kommando `ls(1)` - den Inhalt verschiedener Verzeichnisse ausgeben kann.

Es wird empfohlen beim Entwurf des Programms in folgenden Arbeitschritten vorzugehen:

(a) Schreiben sie zunächst ein Programm, welches alle Einträge des aktuellen Verzeichnisses (`'.'`) als je einen Eintrag pro Zeile ausgibt. Einträge, deren Name mit einem `'.'` beginnen, sollen nicht angezeigt werden (versteckte Dateien). (`opendir(3)`, `readdir(3)`, `closedir(3)`)

(b) Erweitern Sie das Programm nun so, dass vor dem Dateinnamen, auch die Dateigröße ausgegeben wird. Name und Größe sollen durch einen Tabulator (`'\t'`) getrennt werden. Am Ende der Ausgabe soll die Gesamtzahl der ausgegebenen Einträge, sowie deren Gesamtgröße ausgeben werden. Ignorieren Sie hierbei alle Einträge, bei denen es sich nicht um eine reguläre Datei handelt. (`lstat(2)`)

(c) Werten Sie nun die Parameter `argv` aus. Alle übergebenen Parameter sollen als Verzeichnispfade interpretiert werden und wie in (a) und (b) beschrieben ausgegeben werden. Die Ausgabe eines Verzeichnisses soll mit `'<Verzeichnisname>:\n'` beginnen. Wird kein Parameter übergeben, soll das aktuelle Verzeichnis ausgegeben werden.

Hinweise:

- Sie finden im Verzeichnis /proj/i4spic/pub/aufgabe7 für jede Teilaufgabe eine Beispiellösung, deren Ausgabe Sie mit Ihrer eigenen Lösung vergleichen können.
- Für die Abgabe ist nur die endgültige Lösung notwendig, die Sie in einer Datei `printdir.c` in Ihrem Projektverzeichnis ablegen sollen.
- Achten Sie auf aussagekräftige Fehlermeldungen, die alle auf dem Standardfehlerkanal ausgegeben werden sollen.
- Ihr Programm muss nur Pfade und Dateinamen bis zu einer Gesamtlänge von 1024 Zeichen[1] behandeln können. Achten Sie bei zu langen Pfaden und Dateinamen auch hier auf eine entsprechende Fehlermeldung.
- Die Funktionen zur Behandlung von Zeichenketten aus `string.h` sind beim Lösen der Aufgabe hilfreich.
- Übersetzen Sie das Programm mit
  `gcc -pedantic -Wall -Werror -std=c99 -D_XOPEN_SOURCE=500 -O3 -o printdir printdir.c`
- Alternativ können Sie `make` verwenden. Hierzu muss einmal
  `export CFLAGS="-pedantic -Wall -Werror -std=c99 -D_XOPEN_SOURCE=500 -O3"`
  angegeben werden. Anschließend können Sie Ihr Programm mit
  `make printdir`
  übersetzen.

Beispielausgabe

```
$ ./printdir ~/test/first_path ~/test/second_path
/home/alice/test/first_path:
127 file1.txt
157 file2.txt
68  test_dir
2 Dateien; 284 Bytes
/home/alice/test/second_path:
68  dir1
68  dir2
116 fileA.txt
115 fileB.txt
2 Dateien; 231 Bytes
```

Abgabezeitpunkt

T01	28.06.2015	18:00:00
T02	28.06.2015	18:00:00
T03	29.06.2015	18:00:00
T04	29.06.2015	18:00:00
T05	30.06.2015	18:00:00
T06	30.06.2015	18:00:00
T07	30.06.2015	18:00:00
T08	01.07.2015	18:00:00

[1] Alternativ kann auch `PATH_MAX` aus der `limits.h` verwendet werden.