
SPiCAufgabe #6: printdir

(12 Punkte, 11.12.2013 20:00, keine 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 Arbeitsschritten vorzugehen:

- 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)`)
- Erweitern Sie das Programm nun so, dass vor dem Dateinamen, 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 ausgegeben werden. Ignorieren Sie außerdem alle Einträge, bei denen es sich nicht um eine reguläre Datei handelt. (`lstat(2)`)
- Werten Sie nun die Parameter `argv` und `argc` aus. Alle übergebenen Parameter sollen als Verzeichnispfade interpretiert 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/aufgabe6` 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 Zeichen¹ 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. (`string(3)` taugt leider nichts. Google hilft!)
- Übersetzen Sie das Programm mit

```
gcc -pedantic -Wall -Werror -std=c99 -D_BSD_SOURCE -O3 -o printdir printdir.c
```
- Alternativ können Sie `make` verwenden. Hierzu muss einmal

```
export CFLAGS="-pedantic -Wall -Werror -std=c99 -D_BSD_SOURCE -O3"
```

angegeben werden. Anschließend können sie Ihr programm mit

```
make printdir
```

übersetzen.

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