

## Aufgabenblatt 8

### Aufgabe – Dateisysteme

Diese Aufgabe soll die Möglichkeiten des Zugriffs auf spezielle Dateisysteme verdeutlichen. Dabei geht es um das Ihnen bereits bekannte tar-Format, das ursprünglich zur Archivierung von Daten auf Bandlaufwerken entwickelt wurde.

#### Aufgabenstellung:

Implementieren Sie ein C/C++-Programm, das:

1. Für einen als Parameter übergebenen Dateinamen zunächst überprüft, ob es sich um eine Archiv-Datei im *ustar*-Format handelt.
2. In dem Fall, dass es sich bei der Datei um ein solches Archiv handelt, soll der Inhalt des Archivs in folgendem Format (entspricht dem Befehl `tar tvf ...`) ausgegeben werden:  
  

```
drwxr-xr-x gray/gray          0 2019-02-23 13:57 tar-1.32/  
-rw-r--r-- gray/gray        79584 2015-12-05 22:53 tar-1.32/ABOUT-NLS  
-rw-r--r-- gray/gray         2151 2019-01-02 19:07 tar-1.32/TODO  
-rw-r--r-- gray/gray         1207 2019-01-02 19:07 tar-1.32/Makefile.am
```
3. Ein Beispiel-Archiv (gleichzeitig eine Referenzimplementierung des tar-Befehles) ist im Lernraum zur Veranstaltung hinterlegt. Das Archiv ist komprimiert und muss zunächst dekomprimiert werden. Testen Sie dann Ihr Programm mit diesem Archiv und vergleichen Sie die Ausgabe mit der des Unix-tar-Befehls.

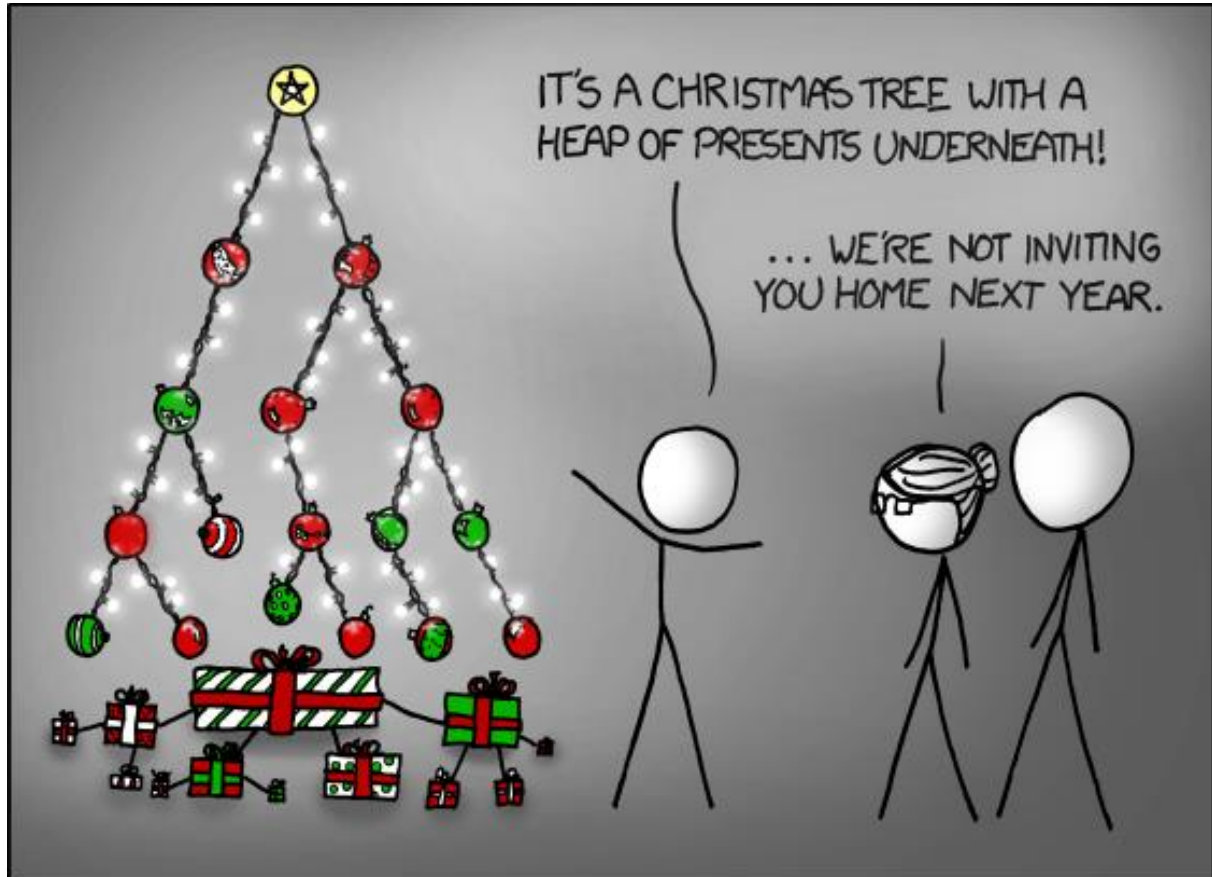
#### Rahmenbedingungen:

- Benutzen Sie für diese Aufgabe also **ausschließlich Low-Level Funktionen** zur Dateibehandlung. Ausgaben auf der Konsole sollen ebenfalls mit diesen Befehlen realisiert werden. Die in Frage kommenden POSIX Befehle sind: **open, close, exit, lseek, read, write, malloc, strdup, free, strptime**.
- Die Standard-Bibliotheksfunktionen wie z. B. **fopen, fprintf, printf, sprintf, put** usw. dürfen in Ihrem Programm **ausschließlich** zu Debugzwecken benutzt werden.
- Fehler (z.B. die Datei kann nicht geöffnet werden) sollen auf **stderr** ausgegeben werden.

#### Hinweise:

- Daten und Meta-Daten (*Header*) werden in einem ustar-Archiv in Blöcken (512 Byte) gespeichert. Die Header-Blöcke müssen Sie der Reihe nach auslesen und interpretieren.
- Beachten Sie, dass Zahlen im Header als *Oktalwerte* dargestellt werden.
- Leere Blöcke können ignoriert werden.
- Die Definition des ustar-Formats ist analog zu den Systembefehlen zum Dateizugriff / -manipulation in einem POSIX-Standard festgelegt. Eine für die Bearbeitung der Aufgabe hinreichende und kompakte Beschreibung des Formates finden Sie in der Wikipedia.
- Bei der Analyse der binären Dateien (z. B. im tar-Format) kann das Linux-Kommando **xxd** nützlich sein.

Für das Testat ist ein **Protokoll** (Formatierung: siehe Blatt 1) des erstellten Programms (denken Sie an hinreichende Kommentierung) inklusive der durchgeführten Tests und Antworten auf die gestellten Fragen vorzulegen.



Quelle: <http://xkcd.com/>