

Patrick Felschen, Matr. Nr. 932056
Julian Voß, Matr. Nr. 934505

Aufgabenblatt 8

Implementation

Öffnen

```
int fd = open(argv[1], O_RDONLY);  
if (errno != 0) {  
    printError(strerror(errno));  
    return EXIT_FAILURE;  
}
```

Die Funktion *open* liest eine übergebene Datei mit einem entsprechenden Mode (hier read-only) ein und liefert einen Deskriptor der geöffneten Datei zurück.

Lesen

```
struct TARHeader {                /* offset */  
    char name[100];                /* 0 */  
    char mode[8];                  /* 100 */  
    char uid[8];                   /* 108 */  
    char gid[8];                   /* 116 */  
    char size[12];                 /* 124 */  
    char mtime[12];                /* 136 */  
    char chksum[8];                /* 148 */  
    char typeflag;                 /* 156 */  
    char linkname[100];            /* 157 */  
    char magic[6];                 /* 257 */  
    char version[2];               /* 263 */  
    char uname[32];                /* 265 */  
    char gname[32];                /* 297 */  
    char devmajor[8];              /* 329 */  
    char devminor[8];              /* 337 */  
    char prefix[155];              /* 345 */  
    char padding[12];              /* 500 */  
};
```

Der Header eines Tar-Archivs besteht aus 512 Byte und enthält viele Informationen über die zugehörige Datei. Es wurde ein Struct erstellt, um die Informationen in einer Variable abspeichern zu können.

```
while (read(fd, &tarBuf, sizeof(tarBuf)) > 0) {  
    // Verarbeitung  
}
```

Durch die *read* Funktion kann aus einem Filedeskriptor eine bestimmte Menge an Daten ausgelesen und in einem Buffer gespeichert werden. Hier wird aus *fd* die Größe des Headers ausgelesen und in *tarBuf* abgespeichert. Dies wird wiederholt ausgeführt, um die Header-Informationen aller Dateien abfragen zu können.

Offset-Berechnen

```
long sizeOct = atol(tarBuf.size);
long sizeDec = octalToDecimal(sizeOct);
long segCount = ceil((double)sizeDec / 512);
long segSize = segCount * 512;
long modeOct = atol(tarBuf.mode);
long modeDec = octalToDecimal(modeOct);
long mtimeOct = atol(tarBuf.mtime);
time_t mTimeDec = octalToDecimal(mtimeOct);
```

```
lseek(fd, segSize, SEEK_CUR);
```

Die Größen, die Berechtigungen und die Zeitangaben der einzelnen Dateien des Tar-Archivs sind jeweils im Oktalzahlenformat angegeben. Diese werden mit der *octalToDecimal*-Funktion umgerechnet.

Damit die Dateien nicht in 512 Byte-Schritten durchgegangen werden, wird die Anzahl an 512 Byte-Blöcken bis zum nächsten Header berechnet. Mit dem Ergebnis lässt sich mittels der *lseek*-Funktion der Datei-Zeiger verschieben.

Test

```
/Betriebssysteme/praktikum08 ../praktikum08/dirStruct.tar
drwxr-xr-x  wuebbelm/other    0      2006-09-29 10:49  rootTest/
drwxr-xr-x  wuebbelm/other    0      2006-09-29 10:49  rootTest/dir1/
-rw-r--r--  wuebbelm/other    0      2006-09-29 10:49  rootTest/dir1/file5
[...]
Process finished with exit code 0
```

```
/Betriebssysteme/praktikum08 ../praktikum08/dirStruct.ta
No such file or directory
Process finished with exit code 1
```

```
/Betriebssysteme/praktikum08 ../praktikum08/Blatt_8_Aufgabe.pdf
Not a TAR-Archive.
Process finished with exit code 1
```