



04.09.2017

Sommersemester 2017
Praktikum Betriebssysteme - Tag 1

C-Programmierung – Dateien Kopieren

Das Rahmenprogramm *mycopy.c* für die heutige Übung finden Sie im Ilias.

Aufgabe 1.1 (Datei kopieren)

Schreiben Sie ein C-Programm *mycopy* welches eine Datei kopiert. Der Aufruf soll in der Form

```
mycopy [-b n] src dest
```

erfolgen, wobei *src* und *dest* für die Namen der Ausgangs- und der Zieldatei stehen. Bereits bestehende Zieldateien sollen erst auf Nachfrage überschrieben werden. Mit dem optionalen Flag *b* soll die Größe des verwendeten Puffers angegeben werden können.

Wird keine Puffergröße angegeben, so soll die Größe des Puffers 1 Byte betragen.

Verwenden Sie dazu die UNIX-Systemaufrufe *open(2)*, *read(2)*, *write(2)*, *close(2)* und *stat(2)*. Öffnen Sie die Ausgangsdatei ausschließlich zum Lesen, die Zieldatei ausschließlich zum Schreiben. Weshalb ist dies wichtig? Wie verhält sich Ihr Programm, falls Ausgangs- und Zieldatei identisch sind?

Erstellen Sie eine große Datei im Verzeichnis */tmp*. Benutzen Sie dazu z. B. folgendes Bash-Programm:

```
for ((I=0; $I<50; I++))  
do  
    man gcc >> gcc.man;  
done
```

Messen Sie mit dem Utility *time(1)*, wie lange Ihr Programm zum Kopieren der Datei innerhalb des Verzeichnisses benötigt, wenn Sie *read()* und *write()* einen Puffer von 2^0 Byte, 2^1 Byte, ..., 2^{16} Byte übergeben. (Weshalb sollten Sie die Messungen nicht in Ihrem Homeverzeichnis durchführen?)

Aufgabe 1.2 (Datei umdrehen)

Schreiben Sie ein C-Programm *reverse*, welches eine Datei byteweise umkehrt. Das Programm soll mit der Kommandozeile

```
reverse filename
```

aufgerufen werden und seine Ausgabe in die Datei *filename.rev* schreiben. Das erste Byte in *filename* soll anschließend gleich dem letzten Byte in *filename.rev* sein, das zweite Byte in *filename* gleich dem vorletzten in *filename.rev* und so fort. Verwenden Sie den UNIX Systemaufruf *lseek(2)*.