

4. Übung: Kopieren einer Datei **zwischen zwei Prozessen per Pipe**

Realtime Systems

Abgabe: 23. Juni 2008 (für die Gruppen 1 und 3)
30. Juni 2008 (für die Gruppe 2)

Realisieren Sie das Kopieren einer !beliebigen! Ursprungs-Datei in eine Ziel-Datei dadurch, dass

- ein Prozess (Erzeuger) aus der Ursprungs-Datei jeweils Blöcke der Länge MAXLEN liest und diese in eine Pipe schreibt und
- ein zweiter Prozess (Verbraucher) die Daten aus dieser Pipe liest und sie in die Ziel-Datei schreibt.

Beachten Sie die korrekte Verwendung der File-Deskriptoren zur Benutzung der Pipe durch den Erzeuger- und Verbraucher-Prozess.

Lagern Sie die Zugriffe auf die Ursprungs- und Ziel-Datei in ein separates Quelltext-Modul (z.B. `fileio.c`) aus und gestalten Sie die davon zur Verfügung gestellten Funktionen so einfach wie möglich (z.B. `err = myOpen(quelle, ziel);` und `n = myRead(buffer, len);`... Filedeskriptoren im Modul gekapselt, ...). Die „Dienstleistungen“ dieses Moduls stellen Sie über eine Header-Datei Modul (z.B. `fileio.h`) zur Verfügung.

Das Modul `fileio.c` soll nur die Dateizugriffe (`open`, `close`, `read`, `write` oder `fopen`, ...) enthalten und so vom Mechanismus der Interprozesskommunikation getrennt sein. So können Sie dieses Modul auch für die nächsten Aufgaben unverändert einsetzen. Sinnvoll ist es z.B. auch, dem Nutzer dieses Moduls nur Aufrufe wie z.B. `myOpen(...)`, `myRead(...)`, `myWrite(...)` und `myClose(...)` zur Verfügung zu stellen und die File-Deskriptoren der Quell- und Ziel-Datei im Modul zu kapseln.

Die Namen von Quell und Zieldatei (auch eine absolute Pfadangabe soll möglich sein) sollen beim Aufruf des Programms auf der Kommandozeile angebbbar sein.

Erstellen Sie eine zugehörige Datei `Makefile` zur Verwendung mit dem Hilfsprogramm `make`.

Um das korrekte Kopieren zu überprüfen, können Sie das Kommando `diff` verwenden.