

# Verteilte Systeme

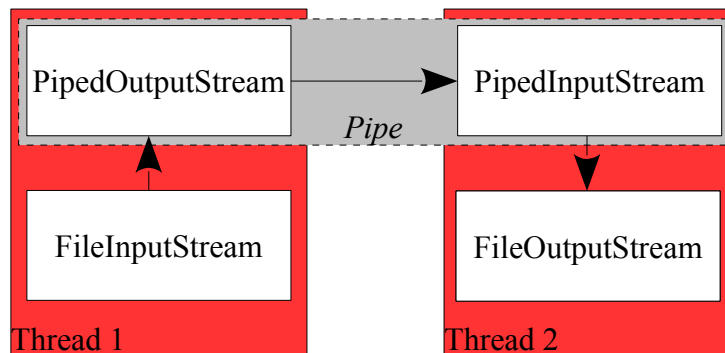
## Übung A1

Bearbeitungszeit 2 Wochen

Diese Übung adressiert Grundlagen zum File-I/O (Java NIO.2) und zu Multi-Threading.

### Aufgabe: Intraprozess-Kommunikation mittels Pipes

Die Klasse *FileCopySingleThreaded* realisiert ein Programm zum Kopieren von Dateien mittels des neuen Java NIO.2 Filesystem-APIs aus Java 1.7. Ersetzt den bisherigen direkten Datentransfer durch zwei Transporter-Threads welche die Daten zum einen von der Quell-Datei in einen *PipedInputStream*, und zum zweiten aus einem *PipedOutputStream* in die Ziel-Datei kopieren.



Kopiert dazu die Klasse *FileCopySingleThreaded* nach *FileCopyMultiThreaded*, und geht folgendermaßen vor:

- Erstellt eine statische innere Klasse *Transporter* welche *Runnable* erweitert, mit den Instanzvariablen *inputStream* und *outputStream*, vom Typ *InputStream* und *OutputStream*. Erstellt einen Konstruktor der beide Instanzvariablen aus Parametern setzt, und schreibe eine *run()*-Methode welche den Inhalt des Quell-Streams in den Ziel-Stream transferiert.
- Ändert die *main()*-Methode der Klasse *FileCopyMultiThreaded* so ab dass sie eine *Transporter*-Instanz (i.e. Synchron, ohne Thread mittels *runnable.run()*!) zum Datentransfer verwendet. Testet bis der Code läuft.
- Ändert die *main()*-Methode der Klasse *FileCopyMultiThreaded* nun so ab dass sie zwei *Transporter*-Instanzen in zwei Threads (i.e. Asynchron, mittels *thread.start()*!), sowie eine *Pipe* zum Datentransfer verwendet. Beachtet dabei dass das Schließen des einen Endes einer *Pipe* automatisch auch das andere Ende schließt! Eure Lösung sollte ohne Maßnahme zur Resynchronisierung terminieren. Testet bis der Code läuft.
- Ein Skalierungserfolg ist durch diese Maßnahme auf normalen Systemen nicht zu erwarten, da nun jedes Byte der zu kopierenden Datei zweimal gelesen und geschrieben werden muss. Auf Systemen mit mehreren Festplatten dagegen kann dies (bei geeignet großer Wahl der *Pipe*-Puffergröße) ganz anders aussehen: Falls Quelle und Ziel auf verschiedenen Festplatten beheimatet sind, kann hier die Reduzierung von Bewegungen der mechanischen Lese/Schreib-Köpfe leicht den zusätzlichen Aufwand beim Datentransfer überwiegen!