

---

## Aufgabe 3: pforward

Entwerfen und programmieren Sie ein Programm **pforward**, welches TCP-Verbindungen an einem lokalen Port entgegennimmt und die Daten an einen beliebigen anderen Port eines anderen Rechners weiterleitet.

Lösen Sie die Aufgabe in folgenden Schritten:

### a) Eine bidirektionale Verbindung weiterleiten

Ihr Programm soll an einem frei wählbaren Port, Verbindungen entgegennehmen (**socket(2)**, **bind(2)**, **listen(2)**, **accept(2)**). Die lokale Portnummer, der Rechnernamen und die Portnummer des Zielrechners werden dem Programm als Argumente übergeben. Nach der Verbindungsannahme soll das Programm eine Verbindungen zum Zielrechner aufbauen (**gethostbyname(3)**, **connect(2)**) und die Daten weiterleiten. Benutzen Sie zum Weiterleiten der Daten die zunächst vorgegebene Funktion

**int forward(int fd1, int fd2).**

Diese Funktion liest von beiden Filedeskriptoren Daten ein und gibt die Daten am jeweils anderen Filedeskriptor wieder aus. Sie finden die benötigte Header-Datei (**forward.h**) und die Objekt-Datei (**forward.o**) im Verzeichnis */proj/i4sos/pub/aufgabe3*. Erstellen Sie ein passendes Makefile.

### b) Die Funktion “forward”

Implementieren Sie die Funktion **forward** nun selbst mit Hilfe von POSIX-Threads (**pthread\_create(3)**). Die Funktion soll in einer eigenen Datei *forward.c* gespeichert werden. Passen Sie ihr Makefile entsprechend an.

### c) Mehrere Verbindungen weiterleiten

Erweitern Sie das Programm nun auf die Verarbeitung von mehreren Verbindungen. Für jede entgegengenommene Verbindung soll ein Sohnprozess erzeugt werden (**fork(2)**), welcher die Daten mit Hilfe der **forward**-Funktion weiterleitet. Der Vaterprozess soll gleich wieder neue Verbindungen entgegennehmen können. Richten Sie einen Signalhandler (**sigaction(2)**) ein, welcher entstehende Zombie-Prozesse aufräumt.

## Hinweis zur Lösung dieser Aufgabe:

- Zum Testen können sie Ihre Programm wie folgt starten:  
pforward <freigewählter port> wwwproxy.informatik.uni-erlangen.de 8080  
und im Webbrowser den lokalen Rechner mit der freigewählten Portnummer als Proxy eintragen.
- Sockets sind nicht Bestandteil des POSIX-Standards. Deshalb müssen Sie Ihr Programm mit der Option `-D_XOPEN_SOURCE=500` übersetzen.
- Die pthread-Funktionen sind in einer speziellen Funktionsbibliothek (libpthread) zusammengefasst, die Sie beim Compilieren bzw. Binden Ihres Programms mit angeben müssen (Option `-lpthread`).