Aufgabe 9:

25.06.2007

port_forward (12 Punkte) Bearbeitung in Zweier-Gruppen

In dieser Aufgabe soll ein weiteres Modul für den Connection Broker entwickelt werden, welches die Daten zwischen der vom Connection Broker aufgebauten Verbindung und einem beliebigen Port eines beliebigen Rechners bidirektional weiterleitet. Alle Vorgaben zu dieser Aufgabe finden Sie wie üblich im Verzeichnis */proj/i4sos/pub/aufgabe9*. Sie finden dort neben einem für die Aufgabe tauglichen Makefile auch x86-Objekt-Dateien aller für diese Aufgabe notwendigen Module. Sie können also beliebige Teile Ihrer Implementierung durch die Vorgabe ersetzen und die Funktionalität testen. Kopieren Sie zunächst alle Vorgaben in Ihr Projektverzeichnis und lösen Sie die Aufgabe dann in folgenden Schritten:

a) Anpassung des Connection Broker

Der Rechnername und die Portnummer für den zweiten Endpunkt sollen dem port_forward Modul als Kommandozeilenparameter übergeben werden. Die in Aufgabe 8 angebotene Schnittstelle zwischen Connection Broker und Dienstleistermodulen bietet diese Möglichkeit nicht und wurde daher erweitert. Die neue Signatur der **new_client()**-Funktion finden Sie inkl. Beschreibung in der aktualisierten Vorgabe **client-handler.h**. Die Parameter für das Modul werden beim Aufruf des Connections Brokers durch "--" vom Parameter des Connection Brokers abgetrennt, der Aufruf für **port_forward** sieht also folgendermaßen aus:

./port_forward <ConnBrokerPort> -- <Ziel-Rechner> <Ziel-Port>

Sie können Ihre Connection Broker- und printtime-Module aus Aufgabe 8 an die neue Schnittstelle anpassen. Hierbei sollen die Modulparameter aber optional sein, d.h. der Aufruf des Connection Brokers soll nach wie vor wie in Aufgabe 8 funktionieren. Alternativ können Sie das vorgegebene **connbroker.o** verwenden. Die Anpassung des Connection Broker-Moduls ist optional und geht nicht in die Bewertung der Aufgabe ein.

b) Verbindung bidirektional weiterleiten (Modul port-forward)

Implementieren Sie in einer Datei **port_forward.c** ein Connection Broker Service-Modul, welches den Rechnernamen und die Portnummer des Zielrechners als Argumente übergeben bekommt. Das Modul soll eine Verbindung 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. In der Vorgabe finden Sie die benötigte Header-Datei (**forward.h**) und Objekt-Dateien (**forward-select.o, forward.o**) mit zwei alternativen Implementierungen.

c) Das Modul "forward"

Implementieren Sie das Modul **forward** nun selbst mit Hilfe von POSIX-Threads (**pthread_create(3)**) in einer Datei **forward.c**. Es soll jeweils ein Thread die Weiterleitung der Daten in eine Richtung übernehmen.

Hinweise zur Lösung dieser Aufgabe:

- Zum Testen können Sie Ihr Programm wie folgt starten:
  ./port_forward <freigewählter port (>1024)> -- wwwproxy.informatik.uni-erlangen.de 8080
  und im Webbrowser den lokalen Rechner mit der freigewählten Portnummer als Proxy eintragen.
- Alternativ können Sie auch einen Port zu Ihrem timed weiterleiten und mit **nc(1)** zu der freigewählten Port-Nummer statt der Port-Nummer des timed verbinden.
- 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 speziallen Funktionsbibliothek (libpthread) zusammengefasst, die Sie beim Kompilieren bzw. Binden Ihres Programms mit angeben müssen
  (Option **`-lpthread`**).
- Beim Weiterleiten der Verbindung zum Webproxy-Server kommt es immer wieder zu Problemen mit unvollständig geladenen Seiten. Für ein korrektes Funktionieren der Weiterleitung ist es erforderlich, dass die forward Funktion erst nach dem Ende **beider** Threads zurückkehrt.
- In den Modulen forward und port_forward soll keinesfalls der Prozess beendet werden. Wie in Aufgabe 8 gilt, dass die new-client() Funktion immer zurückkehren muss. Im Fehlerfall sollen die beiden Dienstmodule die Fehlermeldung auf den Standardfehlerkanal ausgeben und den Fehler mit dem Rückgabewert -1 signalisieren.

Abgabe: bis spätestens Donnerstag, 05.07.2007, 12:00 Uhr