

Projekt Systementwicklung SS 2024 - Aufgabe 1 (Stand 18.03.24)

Vorbereitungen für das Praktikum:

Lesen Sie den HTTP 1.0 Standard (RFC 1945). Sie müssen zum Beginn des Praktikums die Terminologie (Abschnitt 1.2) von HTTP 1.0 grundlegend verstehen und HTTP Requests (besonders Abschnitte 5.1.1, 5.1.2, 5.2) und HTTP Responses (besonders Abschnitte 4, 6, 9) verstehen.

Lesen und verstehen Sie weiterhin die Praktikumsaufgaben 1 bis 3 und überlegen Sie sich, welche Schritte nötig sind, um die Praktikumsaufgaben zu lösen.

Implementieren Sie die `str_cat`-Funktion in der HTTP-Lib und testen Sie die korrekte Funktionalität. Sie müssen im Praktikum diese Funktion zeigen und Ihre Lösung detailliert erklären können. Bereiten Sie diese Funktion und entsprechende Tests daher sorgfältig vor.

Aufgabe a) Projektplan

Erstellen Sie bis zum Ende des Praktikums eine Liste der zu erledigenden Arbeitspakete („Projektplan“) aus den Aufgaben 2 und 3. Pro Praktikumsaufgabe ist es durchaus angebracht, 10-20 Arbeitspakete aufzulisten. Verteilen Sie die Arbeitspakete dann auf die Gruppenmitglieder und versehen Sie sie jeweils mit einer Deadline, zu welcher das jeweilige Arbeitspaket fertiggestellt werden muss. Hier ein Beispiel:

Pos.	Beschreibung	Verantwortlicher	Deadline
1	Herauslesen des HTTP-Pfads aus dem eingehenden HTTP-Request. Als Ergebnis wird das relevante Feld in der Struktur <code>http_request</code> ausgefüllt.	Martina Mustermann	02.04.
2	Erstellen einer Funktion, die kodierte Zeichen wie <code>%20</code> aus einem Pfad in das korrekte Zeichen umwandelt und den neuen Pfad zurückgibt.	Christian Müller	05.04.
3

Wählen Sie weiterhin eine Projektleiter*in, die den Projektfortschritt überwacht. Es liegt in der Verantwortung der Projektleiter*in während des gesamten Kurses dafür zu sorgen, dass die Arbeitspakete zeitgemäß fertig gestellt werden. Ist es absehbar, dass das verantwortliche Gruppenmitglied das Arbeitspaket nicht zeitgemäß fertig stellt, dann muss die Projektleiter*in dieses Arbeitspaket entsprechend umplanen.

Sprechen Sie frühzeitig mit den Tutoren, wenn absehbar wird, dass Gruppenmitglieder auch nach mehrfacher Aufforderung nicht den gestellten Aufgaben nachkommen.

Aufgabe b) HTTP-Lib

Um einen HTTP-Server in C entwickeln zu können, müssen Sie sich sehr intensiv mit String-Verarbeitung befassen. Leider sind die meisten der in der Standard-Libc verfügbaren Bibliotheksfunktionen dafür nicht oder nur unzureichend geeignet. Genau genommen sind Funktionen, die mit Null-terminierten Strings arbeiten (`strcpy`, `strcat`,

strncpy, strlen, ...) für diesen Kurs gänzlich ungeeignet! Kapseln Sie daher in den folgenden Praktika etwaige Helferfunktionen, die sie selber entwickeln und die Strings ohne Terminator-Bytes verwenden. Schauen Sie sich dafür den Code in den Dateien `httplib.h` und `httplib.c` an.

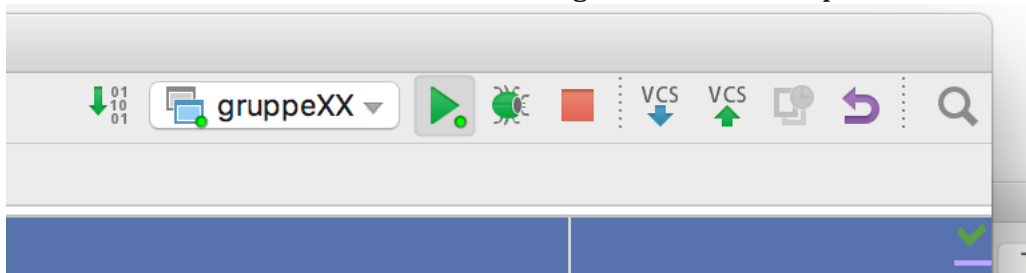
Aufgabe c) Arbeitsplatz einrichten

Sie finden im Ilias Videos, die beschreiben, wie Sie Ihr Projekt in Clion erstellen und ins Gitlab übertragen. Führen Sie weiterhin die folgenden Schritte durch:

1. Benennen Sie in Ihrem Projekt die `main.c` um in `echo_server.c`.
2. Kopieren Sie den Quellcode des Echo-Servers in die `echo_server.c`. Sie können den Quellcode hier herunterladen:
https://git.fh-muenster.de/schinzelpse/-/blob/main/echo_server.c
Kopieren Sie auch die `httplib.h` und `httplib.c` in das Projekt.
3. Comitten und Pushen Sie Ihre Änderungen ins Repository. Wählen Sie dafür „VCS“ → „Commit Changes“. Schreiben Sie eine aussagekräftige „Commit Message“ und wählen Sie „Commit and Push“.
4. Überprüfen Sie, dass die Daten erfolgreich ins Gitlab übertragen wurden.

Aufgabe d) Den Echo-Server ausführen

1. Starten Sie den Server, indem Sie auf den grünen „Run“-Knopf klicken.



Den Echo-Server testen

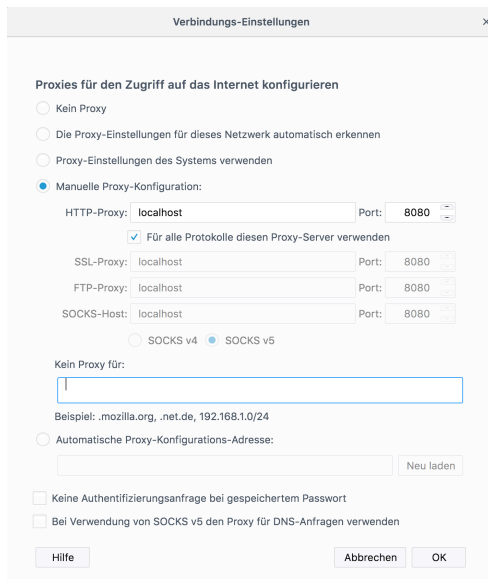
1. Öffnen Sie ein Terminal und geben Sie den Befehl `nc localhost 31337` an.
2. Geben Sie jetzt etwas Text ein und bestätigen Sie.
3. Der Server sollte den von Ihnen eingegebenen Text empfangen und wieder auf der Kommandozeile ausgeben.

Testen Sie den Echo-Server jetzt mit einem Web-Browser

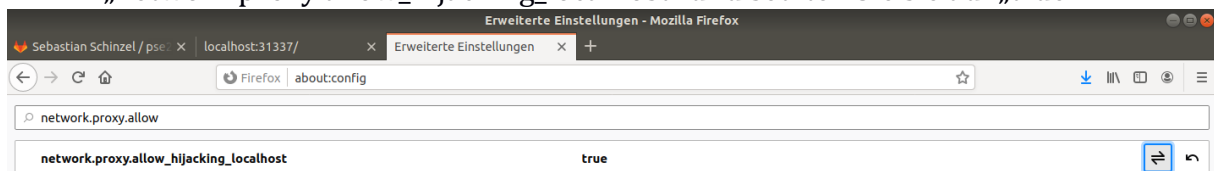
1. Starten Sie Firefox und besuchen Sie die URL `http://localhost:31337/`
2. Der Browser zeigt jetzt etwas Text an.
3. Was bedeutet dieser Text?

Installieren Sie den HTTP-Proxy

1. Laden Sie die burp-Suite von der URL <https://portswigger.net/burp/freedownload> herunter. (Auf der PSE-VM ist sie bereits installiert.)
2. Starten Sie den Burp-Proxy im Start-Menü.
Machen Sie sich weiterhin vertraut mit der Konfiguration von Firefox und richten Sie ein, dass der burp-Proxy für alle Anfragen verwendet wird (Menü → „Einstellungen“ → „Netzwerk-Proxy“). Die Konfiguration sollte wie folgt aussehen:



In aktuellen Versionen von Firefox benötigt man noch eine weitere Einstellung. Besuchen Sie die Webseite „about:config“ und suchen Sie die Direktive „network.proxy.allow_hijacking_localhost“ und setzen Sie sie auf „true“:



3. In burp, wechseln Sie auf dem Reiter „Proxy“ und schalten Sie „Intercept is on“ auf „Intercept is off“
4. Stellen Sie jetzt in Firefox einige Anfragen an den Echo-Server.
5. Wechseln Sie in burp auf den Reiter „HTTP History“ und machen Sie sich mit den HTTP-Anfragen („Requests“) und die Antworten („Responses“) vertraut.
6. Wechseln Sie jetzt wieder in den Modus „Intercept is on“ und setzen Sie im Browser wieder eine Anfrage ab.
7. Ändern Sie den User-Agent in der HTTP-Anfrage von „Firefox“ auf „FH-Muenster-Browser 1.0“ und drücken Sie auf „Forward“.
8. Zeigen Sie, dass am Echo-Server tatsächlich der veränderte User-Agent ankam.

Ändern von Werten innerhalb des Debuggers

1. Stoppen Sie die laufende Instanz des EchoServers, indem Sie im Clion auf das kleine rote Quadrat klicken.
2. Setzen Sie einen Breakpoint in Zeile 237, indem Sie rechts neben die Zeilennummer klicken. Ein roter Punkt erscheint jetzt rechts neben der Zeilennummer.
3. Starten Sie den Server im Debugger, indem Sie auf das kleine „Käfer“-Icon klicken und starten Sie die Ausführung mit einem Klick auf den grünen Pfeil („Resume Program“).
4. Öffnen Sie ein Terminal und geben Sie den Befehl `nc localhost 31337` an. Geben Sie jetzt den Text `Hallo Welt!` ein und bestätigen Sie.
5. Im „Variables“-Fenster ändern Sie den Inhalt der Variablen `response->str` auf einen anderen Wert, z. B. `"Moin!"`.
6. Zeigen Sie, dass der Echo-Server tatsächlich den veränderten Wert zurückgegeben hat.
7. Zeigt der Browser neben Ihrer veränderten Eingabe noch einige Bytes an, die „Datenmüll“ enthalten? Wenn ja, können Sie dieses Verhalten erklären?

Abnahme am Ende Ihres ersten Praktikum-Termins

1. Aufgabe a) Legen Sie den Projektplan Ihrer Gruppe vor.
2. Aufgabe b) Zeigen Sie Ihre eigene Implementierung der `str_cat`-Funktion für die HTTP-Lib.
3. Aufgabe c) Zeigen Sie, dass Sie die Anfragen im burp-Proxy ändern können.
4. Aufgabe c) Zeigen Sie, dass Sie die Antworten im Clion-Debugger ändern können.