Portfolioprüfung – Werkstück A – Alternative 4

1) Aufgabe

Entwickeln und implementieren Sie eine **Client-Server-Anwendung** nach folgendem Schema.

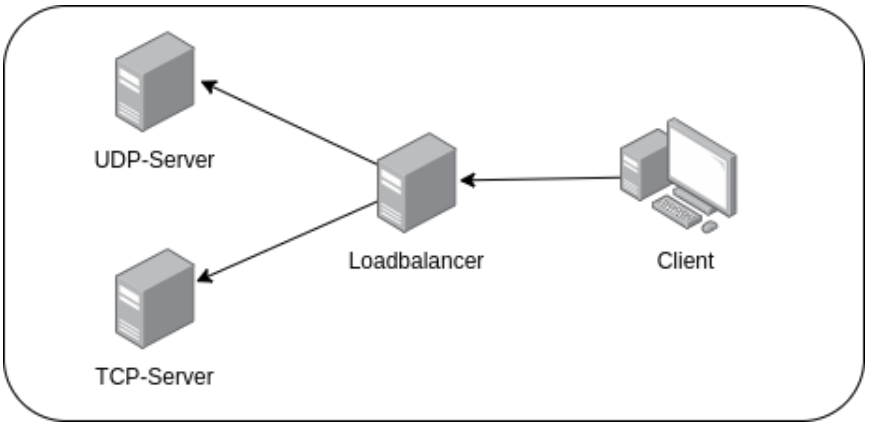

Abbildung 1: Schema Netzwerkaufbau

Entwickeln und implementieren Sie Ihre Anwendung in Python als freie Software (Open Source) und verwenden Sie hierfür ein Code-Repository, z.B. bei GitHub oder GitLab (siehe Abschnitt 4).

Die Anwendung besteht aus vier Prozessen:

- Ein Prozess für den Loadbalancer
- Zwei Prozesse für die Server
- Ein Prozess für den Client

Innerhalb Ihrer Anwendung sollen der Client und Server über das Netzwerk miteinander kommunizieren und Nachrichten austauschen. Implementieren Sie Sockets für die Kommunikation mit den Protokollen TCP (Transport Control Protocol) und UDP (User Datagram Protocol) im Loadbalancer. Dieser leitet je nach Socket die Anfrage zum entsprechenden Server weiter. Der HTTP-fähige Server reagiert auf GET, POST und DELETE und sendet dem Client eine entsprechende Antwort.

In Ihrer Anwendung soll es dem Nutzer möglich sein, die Adresse des Servers über die Kommandozeile einzugeben und eine Nachricht an den Server zu senden. Dazu soll Ihre Anwendung die Systembibliothek `socket` benutzen. Ihre Anwendung soll die eingegebene IP-Adresse mit der Funktion `gethostbyname()` in einen Hostnamen umwandeln. Weiter soll Ihre Anwendung dem Nutzer ermöglichen, via die Funktion `getservbyname()` Service-Informationen zu erhalten und zu nutzen. Nutzen Sie ein

Modul wie zum Beispiel `pypcap` oder `python-libpcap` um Pakete mitzuschneiden und diese in einer Datei persistent zu speichern. Speichern Sie auch die ankommenden Informationen in einem Server-Logfile.

Bearbeiten Sie die Aufgabe in Teams zu maximal **4 Personen** oder **5 Personen** (siehe hierzu Abschnitt 3).

Schreiben Sie eine aussagekräftige und ansehnliche **Dokumentation** (Umfang: **8-10 Seiten**) über Ihre Lösung. Eine Vorlage für das Textsatzsystem LaTeX und weitere Informationen zum Verfassen einer guten Dokumentation finden Sie auf der Vorlesungsseite. Sie müssen die Vorlage nicht zwingend verwenden und Sie müssen auch nicht zwingend LaTeX verwenden. Dieses Projekt ist aber eine sehr gute Gelegenheit sich mit LaTeX auseinander zu setzen und die Vorlage enthält viele hilfreiche Hinweise. In der Dokumentation beschreiben Sie Ihre Lösung (Architektur, etc.), die Aufteilung der Komponenten auf die im Team beteiligten Personen, ausgewählte Probleme und deren Lösungen, etc. Auch einzelne Screenshots können sinnvoll sein. Die Dokumentation soll ein technischer Bericht sein, in der dritten Person formuliert sein, und ganz auf relevante Aspekte Ihrer Lösung fokussieren.

Bereiten Sie einen **Vortrag mit Präsentationsfolien und Live-Demonstration** (Umfang: **15-20 Minuten**) vor. Demonstrieren Sie die Funktionalität der Lösung in der Übung. Eine Vorlage für Präsentationsfolien für das Textsatzsystem LaTeX mit der Erweiterungsklasse `beamer` und weitere Informationen zum Verfassen einer guten Projektpräsentation finden Sie auf der Vorlesungsseite. Auch diese Vorlage müssen Sie nicht zwingend verwenden und Sie müssen auch für die Präsentationsfolien nicht zwingend LaTeX verwenden. Dieses Projekt ist aber eine sehr gute Gelegenheit sich mit LaTeX und der Erweiterungsklasse `beamer` auseinander zu setzen und die Vorlage enthält viele hilfreiche Hinweise. Der Vortrag fokussiert ganz auf Ihre Lösung und nicht auf die Aufgabe oder Inhalte der Vorlesung. Es geht hierbei ausschließlich um Ihre Leistung im Projekt. Es versteht sich von selbst, dass alle im Team beteiligten Personen, ihre selbst entwickelten und implementierten Komponenten vorstellen und demonstrieren können und natürlich auch Fragen zur Lösung und zu ihren Komponenten beantworten können.

2) Anforderungen an die Client-Server-Anwendung

- Der Quellcode soll durch Kommentare verständlich sein.
- Die Anwendung soll an passenden Stellen über ein Fehlerhandling verfügen.
- Alle Komponenten der Anwendung sollen Kommandozeilenanwendung sein.
- Client:
  - Benutzer sollen zur Laufzeit des Clients eingeben können auf welchen Server sie über den Loadbalancer zugreifen wollen und welche Payload mit welcher HTTP-Methode übertragen wird. Beispiel-Prompt in `client.py`:

    ```
    Connect to: TCP-Server
    Message: Hello World!
    Method: PUT
    ```

  - Mit den abgefragten Informationen wird eine vollständige Payload zusammengebaut.
  - Nutzen Sie das Modul `socket`.
  - Ausgabe der Antworten von den Servern auf dem Terminal des `client.py`.
- Loadbalancer:
  - Unterscheidet anhand des Ports an welchen Server die Anfrage weitergeleitet werden soll.
  - Nutzen Sie die Module `socket` und `threading`.
- UDP-Server:
  - Ausgabe aller relevanten Informationen im Terminal des `udp-server.py`
  - Nutzen Sie das Modul `socket`.
- TCP Server:
  - Ausgabe aller relevanten Informationen im Terminal des `tcp-server.py`
  - Nutzen Sie zur Umsetzung das Modul `http.server`.
- Benutzer sollen beim Start eines (TCP oder UDP) Servers den Pfad und Dateinamen der Logdatei als Kommandozeilenargument frei definieren, also z.B. `-logdatei <dateiname>`.

3) Erweiterung der Aufgabe für 5 Personen

- Implementieren Sie einen Netzwerksniffer als weiteres Programm. Dieser hat folgende Anforderungen:
  - Speicherung aller relevanten Protokolldaten.
  - Verwendung des Moduls `python-libpcap` oder einer vergleichbaren Lösung.
  - Persistente Speicherung der Ergebnisse in menschenlesbarer Struktur.
  - Fähigkeit, die gespeicherten Daten mit dem Programm selbst auszulesen.
- Implementieren Sie einen Server, welcher TLS implementiert und vergleichen Sie die Ergebnisse der Netzwerkaufzeichnung mit und ohne TLS.

4) Einige abschließende Worte zum Code-Repository

Das Code-Repository müssen alle am Team beteiligten Personen erkennbar verwenden, das heißt sie müssen ihre Komponente(n) aktiv entwickeln. Die aktive Mitarbeit bei der Entwicklung und Implementierung des Programms muss für alle Teammitglieder in der Commit-Historie des Code-Repositories erkennbar sein. Kommen einzelne am Team beteiligten Personen nicht in der Commit-Historie des Code-Repositories vor, ist deren Beitrag bei der Implementierung des Programms mehr als unglaubwürdig.

Das Code-Repository soll während der gesamten Projektlaufzeit einsehbar und die kontinuierliche Entwicklung des Simulators erkennbar sein. Stellen Sie hierfür Ihr Repository auf Public und nicht auf Private. Ein oder nur sehr wenige Commits einer einzelnen Person am Ende der Projektlaufzeit sprechen gegen eine gemeinsame Entwicklung und sinnvolle Teamarbeit.

5) Literatur

- Foliensätze 7 bis 12 der Vorlesung **Betriebssysteme und Rechnernetze** im SS2024
- **Computer Netzwerke kompakt**, *Christian Baun*, 6. Auflage, Springer Vieweg (2022)
- **Linux-UNIX-Programmierung**, *Jürgen Wolf*, 2. Auflage, Rheinwerk Computing (2006)
- **Offizielle Webseite von `python-libpcap`**, `https://python-libpcap.readthedocs.io/en/latest/`
- **Offizielle Webseite von `socket`**, `https://docs.python.org/3/library/socket.html`
- **Offizielle Webseite von `threading`**, `https://docs.python.org/3/library/threading.html`
- **Offizielle Webseite von `http.server`**, `https://docs.python.org/3/library/http.server.html`