

Programmieren einer Webseite auf der man C++-Files schreiben und ausführen kann

Nicola Krull

Inhaltsverzeichnis

1	Abstract	2
2	Einleitung	2
3	Aufbau	3
3.1	Design	3
3.2	Software	3
4	Security	4
4.1	Sicherheitsprobleme	4
4.2	Mögliche Sicherheitsansätze	4
4.2.1	chroot	4
4.2.2	Containervirtualisierung	4
4.2.3	Hypervisor	5
4.3	Angewendeter Ansatz	6
5	Textfunktionen	7
5.1	Syntax-Highlighting	7
5.2	andere	7
6	Erweiterungsmöglichkeiten	8

1 Abstract

2 Einleitung

3 Aufbau

3.1 Design

3.2 Software

Das Ziel des Projektes ist eine dynamische Webseite zu bauen. Eine dynamische Webseite ist eine Webseite bei welcher der Server mit der Webseite kommuniziert.[1] Für den Bau einer dynamischen Webseite benötigt man ein Webframework, welches die Interaktionen zwischen den Files steuern kann. Dafür wird das Webframework *Express* verwendet. Es ist ein serverseitiges Webframework, welches für die Plattform *Node.js* entwickelt wurde.[2] *Node.js* ist eine open-source, serverseitige Plattform, welche *JavaScript* als Skriptsprache verwendet.[3] Wenn man eine dynamische Webseite programmiert muss man immer zwischen serverseitigen und clientseitigen Code unterscheiden. Für die serverseitigen Programme verwendet man, die Plattform *Node.js* und auf der Clientseite läuft *JavaScript*, *HTML* und *CSS*-Code. Dabei wird in diesem Fall die Templatesprache *Jade* verwendet, welche zur Generierung von *HTML*-Seiten zuständig ist. Es vereinfacht nicht nur die Syntax, sondern es funktioniert wie anderen Programmiersprachen, somit kann man für den *HTML*-Code Variablen, If-Abfragen und for-Schleifen benutzen. [4]

Express kann man in die drei Teile, views, routes und controllers aufteilen. In dem views-Ordner sind die Jade Files abgeschrieben. Diese Files sind für das Design der Webseite zuständig. Also wenn man wie zum Beispiel eine Textbox und einen Button auf der Webseite sehen möchte, müsste man in einem *Jade*-file eine Textarea und einen Button kreieren. Durch das coden dieser Elemente hat man eine Textbox und einen Button auf der Webseite aber es würde keine reaktion geben wenn man in die

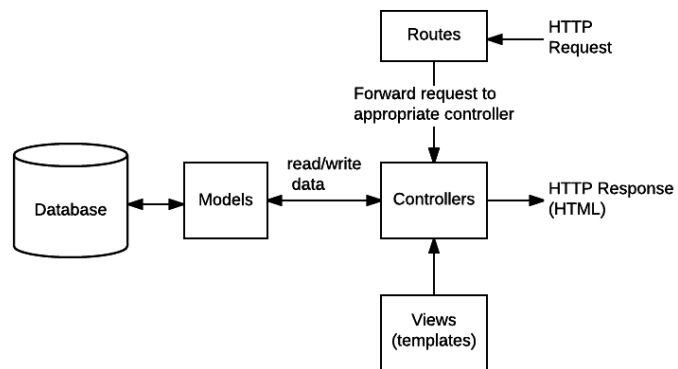


Abbildung 1: Express Aufbau [5]

Textbox hinein schreibt oder den Button drückt. Man möchte den Textinhalt weiterleiten. Dafür benötigt man die HTTP Methoden wie zum Beispiel GET und POST. Die Abkürzung HTTP steht für Hypertext Transfer Protokoll und ist zuständig für die Kommunikation zwischen dem Client und dem Server.[6] All diese HTTP Request werden im routes Ordner bearbeitet und zum Controllers-file weitergeleitet. In den Controllers-Files findet alle Actionen statt. Alles was vom Server berechnet oder ausgeführt werden muss. In meinem Fall wäre das das Compilieren von den Textdaten in der Textbox. Dafür hat man Child Processes von *Node.js* benutzt. Mit Child Processes können die Terminalbefehle automatisch ausgeführt werden und somit muss man es nicht manuell eintippen. Da man keine Datenbanken für meine Arbeit verwende, ist der gebrauch von Models nicht nötig.

4 Security

4.1 Sicherheitsprobleme

4.2 Mögliche Sicherheitsansätze

4.2.1 chroot

Chroot steht für "change root" und es ist ein Programm welches das Rootverzeichnis für Unixsysteme ändern kann.¹ Unix ist ein Betriebssystem, welches in den sechziger Jahren entwickelt wurde. Viele Betriebssysteme basieren auf diesem System, unter anderem das macOS, das iOS und die Linux Betriebssysteme, dazu gehört auch das Betriebssystem Android.² Es generiert eine geschlossene Umgebung namens chroot jail. Diese Umgebung erlaubt den Zugriff auf Files und Befehle ausserhalb dieses Ordners nicht. Somit kann der Users nur auf diesem bestimmten Bereich des Servers zugreifen und somit keinen Schaden an dem Server anrichten.³

4.2.2 Containervirtualisierung

Containervirtualisierung ist ein Verfahren, welches sich auf eine Betriebssystemsfunktion bezieht, bei der der Kernel die Erstellung von multiplen isolierenden User-Space Instanzen erlaubt. Diese erstellten Instanzen werden Containers genannt.⁴ Ein Kernel ist die tiefste Softwareschicht eines Betriebssystems. Der Kernel ist zuständig für die Prozess- und Datenorganisation. Ausserdem kann der Kernel direkt auf die Hardware zugreifen. Die virtuelle Speicherverwaltung wird in User-Space und Kernel-Space unterteilt. Der User-Space ist der Ort an dem die Anwendungssoftwares ausgeführt werden.⁵ Der Unterschied zwischen einem Programm welches in einem Container und eines das von einem Betriebssystem ausgeführt wurde ist, dass das Programm vom OS aus alle Elemente des Betriebssystems zur Verfügung hat, während das File im Container nur auf die Informationen innerhalb des Containers zugreifen kann. Da man komplett isoliert ist muss man einige Elemente wie zum Beispiel Libraries für die Programme, die in diesem Container ausgeführt werden, hinzufügen. Für unixartige Systeme sind Containers fortgeschrittene Implementierungen von chroot.⁶

Eine Containervirtualisierungstechnologie ist Linux Containers (Abkürzung LXC). Es ist ein Verfahren, welches eine Virtualisierung von Softwares auf Betriebssystemebene innerhalb des Linux-Kernels generiert. Das Besondere an den Linux Containers ist, dass sie im Vergleich zu herkömmlichen Virtuellen Maschinen, wie zum Beispiel VMWare oder KVM, einzelne Anwendung in virtuellen Umgebungen ausführen können. Ausserdem ist es möglich ein ganzes Betriebssystem in einem solchen Container zu starten.⁷

¹ <https://en.wikipedia.org/wiki/Chroot> 30.12.2018

² <https://de.wikipedia.org/wiki/Unix> 30.12.2018

³ <https://wiki.archlinux.org/index.php/Chroot> 30.12.2018

⁴ https://en.wikipedia.org/wiki/Operating-system-level_virtualization 30.12.2018

⁵ https://en.wikipedia.org/wiki/User_space 31.12.2018

⁶ https://en.wikipedia.org/wiki/Operating-system-level_virtualization 30.12.2018

⁷ <https://www.webhod.de/lxc-und-lxd-was-sind-linux-container> 30.12.2018

4.2.3 Hypervisor

Bei Hypervisoren, auch Virtual Machine Monitor genannt, handelt es sich um eine Software, welche virtuelle Maschinen von einem beliebigen Betriebssystem aus laufen lassen kann. Es gibt zwei Arten von Hypervisoren.⁸

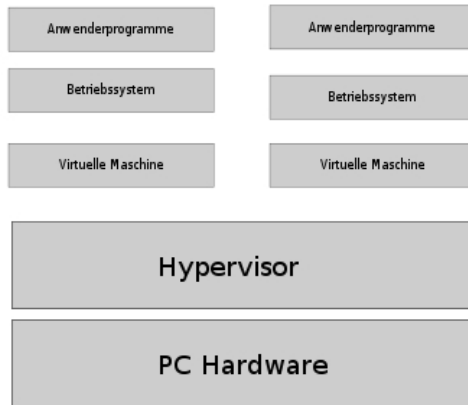


Abbildung 2: Typ-1-Hypervisor

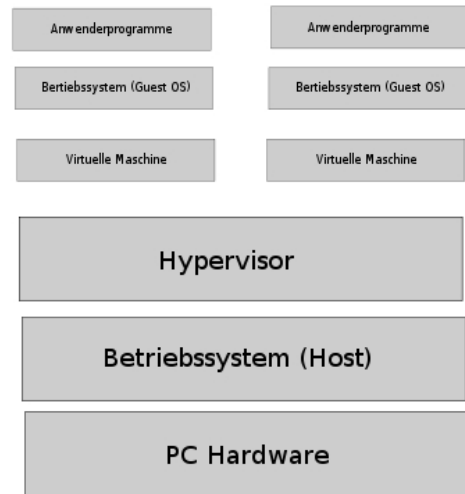


Abbildung 3: Typ-2-Hypervisor

Der Typ-1-Hypervisor auch native oder bare-metal Hypervisor genannt läuft direkt auf der Hardware und benötigt passende Treiber. Dieser Hypervisor benötigt kein vorherige Betriebssystem-Installation. Der Typ-2-Hypervisor braucht ein vollständiges Betriebssystem auf dem Hostsystem. Es nutzt die Gerätetreiber des Benutzersystems für den Zugriff auf die Hardware des Hostsystems.

Das benutzen eines Hypervisor ist ein mögliche Sicherheitsansatz, da er virtuelle Umgebungen kreieren kann. Die Vorteile von diesen virtuellen Umgebungen sind, dass sie vollständig isoliert sind und man kann immer wieder ein weiteres Betriebssystem kreieren oder löschen. Somit würde man ein Shell-Script schreiben, welches automatisch pro User eine virtuelle Umgebung kreiert. In dieser Umgebung würde dann auch die Compilierung stattfinden. Da es in einem virtuellen Betriebssystem stattfinden würde, würde nichts kaputt gehen. Nach dem Kompilieren, würde das Betriebssystem den Output in einen Ordner legen welcher das Hostsystem mit dem virtuellen Betriebssystem teilt. Sobald Output übermittelt wurde, wird das virtuelle Betriebssystem gelöscht.

Ein möglicher Hypervisor ist der KVM. Es steht für Kernel-based Virtual Machine. Der KVM ist seit 2007 in den Linux Betriebssysteme im Haupt-Kernel integriert. Damit es die Virtualisierung unterstützt muss es auf der x86 Hardware installiert sein.

4.2.4

Eine andere Lösung wäre ein System in welchen die Dateiberechtigungen verändert wird und die eingehenden und ausgehenden Ports schliessen würde. In Unix Systeme kann man mit dem Befehl `chmod` die Berechtigungen von Ordnern und Dateien verändern. Dabei gibt es drei Gruppen Eigentümer(user), Gruppe(group) und Sonstige(others). Für jede dieser Gruppen kann man drei verschiedene grund Rechte bestimmen. Das erste Recht `r` steht für den englischen Begriff `read` und bewirkt dass man das Recht hat das File zu lesen. Das zweite Recht wird mit dem Buchstaben `w` ausgedrückt. Es steht für `write` und ermöglicht dem Benutzer Dateien zu schreiben. Das heisst er darf Dateien und Unterverzeichnisse erstellen, löschen, ändern und die Dateirechte der Unterverzeichnisse verändern. Der letzte Buchstabe ist `x` und steht für `execute`. Diese Funktion führt Dateien als Programme aus. Im Falle eines Verzeichnis hätte man die Berechtigung in das Verzeichnis hineinzugehen und so die Dateien und Unterverzeichnisse zu erreichen. Meisst benutzt man eine kombination der drei Berechtigungenfunktionen. Die einzelnen Berechtigungenfunktionen sind zu einer Zahl angeordnet. Das `x` ist gleich Eins, das `w` hat den Wert 2 und das `r` den Wert 4. Wenn man eine kombination der drei Berechtigungen möchte muss man sie addieren.

⁸ <https://www.searchdatacenter.de/definition/Hypervisor-Virtual-Machine-Monitor-VMM> 10.1.2019

4.3 Angewendeter Ansatz

5 Textfunktionen

5.1 Syntax-Highlighting

5.2 andere

6 Erweiterungsmöglichkeiten

Abbildungsverzeichnis

1	Express Aufbau [5]	3
2	Typ-1-Hypervisor	5
3	Typ-2-Hypervisor	5

Literatur

- [1] <https://blog.kompaktdesign.com/webdesign/statisch-vs-dynamisch> 30.12.2018
- [2] <https://de.wikipedia.org/wiki/Express.js> 25.10.2018
- [3] <https://de.wikipedia.org/wiki/Node.js> 25.10.2018
- [4] <https://t3n.de/news/jade-638027/> 25.10.2018
- [5] https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/routes 19.01.2019
- [6] https://www.w3schools.com/tags/ref_httpmethods.asp 25.10.2018