

Fachbeitrag

Andreas Bohne-Lang

Übersichtsplan der PC-Belegung auf der Webseite einer Bibliothek

<http://doi.org/10.1515/abitech-2018-2009>

Zusammenfassung: Wer sich eine einfache Übersicht über belegte und freie PC-Arbeitsplätze in einer Bibliothek verschaffen möchte, kann dies mit wenigen Programmzeilen für seine Bibliothek umsetzen und einen solchen Belegungsplan dynamisch auf einer Webseite darstellen. Ein solcher Belegungsplan, im Eingangsbereich einer Bibliothek dargestellt, hilft den hereinkommenden Benutzerinnen und Benutzern, einen freien PC-Arbeitsplatz zu finden. Ein Erfahrungsbericht.

Schlüsselwörter: Lageplan, Linux, Windows

Overview map of occupied and free computer desks displayed on a library's web page

Abstract: A library may easily offer a simple web page with an overview of occupied and free computer workstations based on the library's ground map with only a few lines of script code. A plan displaying free workstations can be placed in the entrance area of the library to help users find a suitable desk in the library. A report.

Keywords: overview map, Linux, Windows

1 Einleitung

Eine Bibliothek ist weit mehr als ein Medien-Depot mit langen Regalen voller Bücher; sie unterstützt vielmehr die Benutzer und Benutzerinnen aktiv beim Recherchieren, Lernen und Arbeiten auf verschiedenste Weise. Ein solches Serviceangebot ist das Bereitstellen von PC-Arbeitsplätzen, die eine Benutzerin oder ein Benutzer belegen kann. Je nach Größe der Bibliothek ist die prozentuale Fläche der für diesen Arbeitsbereich zur Verfügung gestellten Fläche relativ groß, so dass auch bei kleineren, aber unübersichtlichen Räumen ein Belegungsplan hilfreich wäre, auf dem ein Benutzer oder eine Benutzerin schnell sehen kann, in welchem Bereich noch freie PCs vorhanden sind. Die Grundidee zu dieser Thematik ist nicht neu, und der am KIT in Karlsruhe entwickelte und

inzwischen über eine Firma vertriebene Seatfinder bietet gerade bei großen, verteilten Bibliotheken einen guten Überblick und punktet mit statistischen Auswertungen.

Am KIT in Karlsruhe ging es darum, ca. 1800 Computerarbeitsplätze zu erfassen. Möchte man aber nur eine kleine einfache Übersichtslösung für seine Bibliothek, so kann man dies ohne weiteres mit ein paar Programmzeilen und etwas HTML einfach umsetzen. In diesem kleinen Beitrag wird beschrieben, wie ein solches Projekt für die Bibliothek der Medizinischen Fakultät Heidelberg in Mannheim umgesetzt wurde.

PC-Belegung Bibliothek und Raumbellegung Anbau PC-Pools

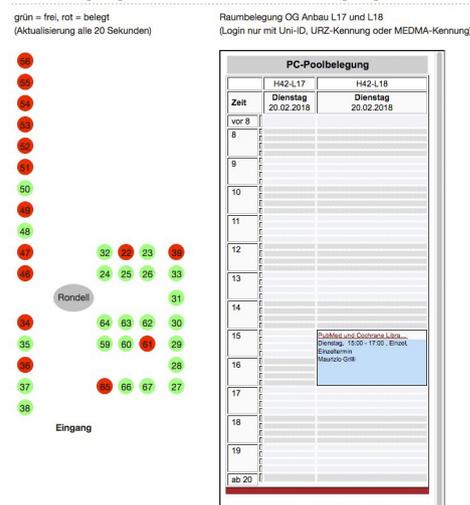


Abb. 1: Anzeige der PC-Belegung auf der Webseite

2 Materialien und Methoden

2.1 Ausstattung

Die Bibliothek stellt ihren Benutzerinnen und Benutzern 38 PC-Arbeitsplätze mit fest installierten Mini-Tower-PCs und Monitoren zum Arbeiten zur Verfügung. Die PCs selber werden unter Windows 7 betrieben und bieten einen Login an zwei Domänen an. Studierende können sich gegen die

Windows-Domäne der Universität Heidelberg authentifizieren, wohingegen die Mitarbeiter und Mitarbeiterinnen sich gegen die Windows-Domäne der Fakultät authentifizieren. Die verschiedenen Web-Server der Fakultät werden unter Linux betrieben. Die Grundidee war, dass man durch ein Skript oder Programm die Windows-PCs remote abfragen kann, ob eine Benutzerin oder ein Benutzer an einem in der Bibliothek zur Verfügung gestellten Computer angemeldet ist oder nicht. Dabei ist es vor dem Hintergrund der Datenschutzbestimmungen wichtig zu betonen, dass es nicht darum geht, herauszufinden, wer angemeldet ist, sondern nur, ob überhaupt an einem bestimmten Arbeitsplatz jemand angemeldet ist – also ein anonymes ja oder nein.

2.1.1 Windows-Abfragen

Würde man eine reine Windows-Umgebung betreiben, in der sowohl die PCs als auch die Web-Server mittels eines Windows-Betriebssystems betrieben würden, so könnte man mit Windows-„Bordmitteln“ eine Abfrage der PCs realisieren. Windows bietet hierzu den query-Befehl an, der mit der Syntax `query user /server:<IP-Adresse des PCs>` abgefragt werden kann. Wichtig ist, dass der Account, der die Abfrage vornimmt, mit Admin-Rechten ausgestattet ist. Alternativ kann man das Programm „PsLoggedOn“ aus den PSTools von Microsoft benutzen. Dieses Paket kann man von der Microsoft-Seite¹ herunterladen und installieren. Eine Abfrage hierzu sähe dann wie folgt aus: `C:\PsTools\psloggedon.exe \\<IP-Adresse des PCs>`. Beide Programme helfen einem allerdings nicht weiter, wenn man eine gemischte IT-Umgebung hat, in der die PCs unter Windows und die Web-Server unter Linux laufen.

2.1.2 Linux-Abfragen

Unter Linux gibt es schon seit langer Zeit das Software-Paket SAMBA, mit dem ein Linux-Rechner Teil eines Windows-Netzwerks werden kann und hier neben Datei- und Druckdiensten auch Controller-Dienste anbieten kann. Ziel des Projektes ist ein nahtloser Übergang zwischen den Systemen (Windows und unixoide Betriebssysteme). Aus diesem Projektpaket kann man das Programm „net“ unter Linux nutzen, um einen Windows-PC remote abzufragen. Hiermit kann man sich die auf dem abgefragten System erzeugte Windows-System-SID (Security Identifier) auflis-

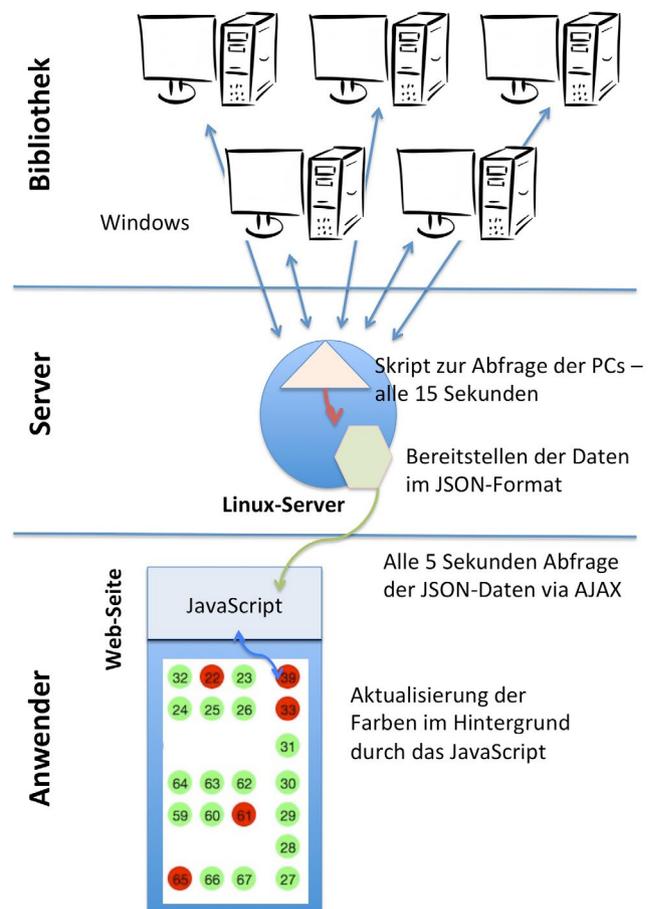


Abb. 2: Datenfluss

ten lassen. Dies erfolgt mit folgendem Befehl unter Linux: `net rpc registry enumerate ,HKEY_USERS' -S <IP-Adresse des PCs> -U Administrator`.

Der in dem Befehl angegebene Administrator kann lokal sein, oder aber auch ein Domänenbenutzer mit Administrator-Rechten. In diesem Fall muss man `-U ad\\Administrator` als Userbezeichnung einsetzen (kleiner Hinweis am Rand: Führt man diesen Befehl unter Linux aus einem Skript aus, so muss man die Backslashes erneut maskieren und man erhält `-U ad\\\\"Administrator'`). Nach der Abfrage des Kennwortes für den angegebenen Benutzer (in unserem Fall der Administrator) bekommt man eine Liste mit Security Identifier-Angaben aufgelistet. Wenn jemand an dem betreffenden Rechner angemeldet ist, taucht in der Liste eine SID mit der Endung „_Classes“ auf. Diese wird als Indikator herangezogen, ob eine Person an dem PC angemeldet ist oder nicht.

¹ Mit einer Suche nach „windows pstools download“ findet man die aktuelle Webseite.

2.2 Skripte

Damit die potentiell vielen Computer effektiv abgefragt werden können, wird in einem ersten Schritt in dem Skript geprüft, ob das Netzwerk-Interface des Rechners aktiv ist oder nicht – also, ob der Rechner eingeschaltet ist. Erst in einem zweiten Schritt wird bei einem eingeschalteten Rechner überprüft, ob jemand angemeldet ist. Damit das Skript effektiv arbeiten kann, werden die Abfragen der verschiedenen Rechner parallel vorgenommen und nicht der Reihe nach, denn hier käme es durch die verschiedenen Time-out-Einstellungen bei ausgeschalteten Rechnern zu einer insgesamt nicht mehr vertretbaren Laufzeit des Skriptes.

2.3 Datenfluss

Das Abfrage-Skript wird als Cron-Job jede Minute mehrmals ausgeführt. Ein Cron-Job ist ein System-Prozess von Linux, der zu speziellen Zeiten automatisiert Aufgaben vornimmt. Das Abfrage-Skript ist in PHP programmiert und wird nicht als Web-Skript, sondern als Programm-Skript ausgeführt. Es dient lediglich dazu, weitere Programme auszuführen und die eingesammelten Daten zu verarbeiten. Eine Übertragung des Verfahrens auf andere Skriptsprachen ist nicht schwer. Das Ergebnis eines Abfragelaufes wird in eine Datei im JSON-Format (JavaScript Object Notation) geschrieben, die aus der Webseite mittels JavaScript gelesen und direkt ausgewertet werden kann. Die Webseite selber stellt als Anzeige für einen belegten PC nur ein div-Element pro Rechner mit dem HTML-Code wie zum Beispiel `<div id="buser42">42</div>` zur Verfügung. Das Aussehen des div-Elementes wird mittels Cascading Style Sheets definiert und im JavaScript-Code dynamisch vorgenommen.

2.4 Farbindikatoren

Insgesamt werden bei der Abfrage der Windows-PCs mehrere Fälle unterschieden, die später auch entsprechend farbig wiedergegeben werden:

- Der PC ist ausgeschaltet oder im Standby-Modus (hellgrün).
- Der PC ist eingeschaltet und kein Benutzer ist angemeldet (dunkelgrün).
- Der PC ist eingeschaltet und ein Benutzer ist angemeldet (rot).
- Der PC ist eingeschaltet, aber eine Remote-Abfrage ist nicht möglich (grau).

3 Ergebnis

Bei der Entwicklung der einzelnen Komponenten der Webseite war ein klar definiertes Ziel des Projektes, dass das Web-Team der Bibliothek bei der Umsetzung die PC-Belegungsseite ohne Programmierkenntnisse in das Layout der Bibliothek integrieren kann. Dies bedeutet, dass man Layout und Skript-Technik strikt voneinander trennen muss. Der JavaScript-Code wird einmal am Anfang der Webseite eingebunden und erledigt alle anfallenden Aufgaben autonom im Hintergrund. Dabei lädt das Skript nach dem Start alle fünf Sekunden dynamisch ohne ein Neuladen der gesamten Seite über AJAX-Technologie (Asynchronous JavaScript and XML) das Ergebnis der PC-Abfrage in die Webseite und aktualisiert diese dynamisch. Das Ergebnis ist somit eine Webseite, auf der mittels einer HTML-Tabelle die in der Bibliothek aufgestellten PCs farbig und sehr aktuell (gemäß der Belegung zum Abfragezeitpunkt) repräsentiert werden.

Die Übersicht in der Webseite orientiert sich am Grundriss der Bibliothek. Da es in einem Anbau noch einen PC-Pool gibt, der vorrangig für Schulungen vorgesehen ist, wird auf der Webseite noch die Kursbelegung aus dem LSF-System (Lehre, Studium, Forschung) tagesaktuell als Auszug eingeblendet. Die Webseite ist responsiv gehalten, so dass auch das Abrufen und Anzeigen auf einem mobilen Endgerät funktioniert. Im Eingangsbereich der Bibliothek wurde ein Rechner mit Monitor sichtbar aufgestellt, so dass jeder hereinkommende Benutzerinnen und Benutzer sich ohne viel Suchen einen Überblick über die PC-Belegung verschaffen kann. Mittels eines QR-Codes der Webseiten-Adresse kann man Benutzern und Benutzerinnen mit mobilen Endgeräten die Eingabe der Web-Adresse erleichtern.



Abb. 3: QR-Code der URL <https://www.umm.uni-heidelberg.de/bibl/pcbelegung>

Downloads

Die Skripte stehen unter <https://github.com/bohnelang/PCBelegung> frei zum Download zur Verfügung.

Autoreninformationen



Dr. Andreas Bohne-Lang

EDV/Bibliothek

Medizinische Fakultät Mannheim

Universität Heidelberg

Ludolf-Krehl-Straße 13–17

68167 Mannheim

bohne-lang@medma.uni-heidelberg.de

orcid.org/0000-0003-4224-735