

SelfLinux-0.10.0



Was ist SelfLinux?

Autor: Heiko Degenhardt (*hede@pingos.org*)
Autor: Matthias Kleine (*kleine_matthias@gmx.de*)
Formatierung: Johnny Graber (*linux@jgraber.ch*)
Lizenz: GFDL

In diesem Kapitel geht es um die Entstehung die Ziele und den aktuellen Stand von SelfLinux.

Inhaltsverzeichnis

1 Geschichte

2 Aneignung von Wissen

3 Derzeitige Linux-Dokumentation

4 Die Ziele von SelfLinux

5 Das Konzept von SelfLinux

5.1 Das Layout des Tutorials


5.2 Die Struktur der Kapitel

5.3 Die Arbeitsweise der Teams



6 Entstehung eines SelfLinux-Dokumentes

7 Ausblick

1 Geschichte

SelfLinux ging 1999 aus dem Projekt  "[PingoS](#)" hervor, das sich zum Ziel gesetzt hat, die Verbreitung von Linux in Schulen zu unterstützen. Noch heute sind einige Teammitglieder auch im PingoS-Projekt aktiv.

Wichtige Eckdaten des Projektes waren:

1999	Gründung des Projektes nach Diskussionen auf der  SuSE Linux Mailingliste
März 2000	Registrierung bei  SourceForge ; damit ergab sich die Möglichkeit, eine professionelle Projektarbeit z.B. unter Verwendung von CVS zu beginnen
April/Mai 2000	Erstellung/Diskussion des ersten "Richtlinienentwurfs" über die Projektarbeit (Verwendung von CVS, Aufgaben der Admins, Bildung von Teams...)
Januar 2001	Neue Datenbank für die Verwaltung der Mitglieder, der Artikel und so weiter
Juli 2001	Erste Testversion auf dem LinuxTag in Stuttgart
Juli 2002	Erstellung eines ersten "Basisrelease" mit den in unseren Augen notwendigsten Texten
September 2002	SelfLinux 0.2.0 released
November 2002	SelfLinux 0.2.1 released
Dezember 2002	SelfLinux 0.4.0 released
Februar 2003	SelfLinux 0.6.0 released
August 2003	SelfLinux 0.8.0 released SelfLinux 0.8.1 released
Oktober 2003	Umzug des CVS-Repositories von Sourceforce zu PingoS

Nach den vielen Diskussionen der Anfangszeit hat sich nun ein Konzept herauskristallisiert, an dem derzeit viele Leute im gesamten deutschsprachigen Raum arbeiten. Die Idee hinter diesem Konzept sowie die möglichen Schritte zur Realisierung sollen im Folgenden beschrieben werden.

2 Aneignung von Wissen

Die Entwicklung der Informationstechnologie schreitet so schnell fort wie kaum eine andere. Die Aneignung des Wissens gerät daher häufig zu einer Art Wissens-Fastfood: Die Zeit ist knapp, aber das fehlende Wissen groß. Daher ist es wichtig, dass der zu lesende Stoff gut aufbereitet präsentiert wird. Denn eine systematische, anschauliche Einführung hilft lernen und Zeit sparen.

3 Derzeitige Linux-Dokumentation

[Linux](#) im engeren Sinne bezeichnet lediglich den Betriebssystem-Kern, mit dem der normale Anwender nur selten unmittelbar konfrontiert wird. In einem weiter gefassten Sinn kann man unter dem Begriff "Linux" jedoch auch all diejenigen Systembestandteile und Werkzeuge zusammenfassen, welche den täglichen Umgang mit einem Linux-System ausmachen und mit welchen der Benutzer oder Administrator aller Wahrscheinlichkeit nach in Kontakt geraten wird.

Für das System im Allgemeinen und auch für die meisten dieser Werkzeuge gibt es heute bereits eine recht gute Dokumentation ([Manual-Pages](#), [Info-Seiten](#), [HTML-Seiten im Web](#), [\(Online-\)Bücher](#), [FAQs](#), [HOWTOs u.v.a.m.](#)). Diese vorhandene Linux-Dokumentation erscheint uns trotzdem ungenügend, denn:

- * Sie besteht aus vielen isolierten Dokumenten.
- * Sie baut folglich nicht logisch aufeinander auf.
- * Sie ist vielfach nur in Englisch verfügbar.
- * Sie ist häufig mangelhaft oder gar nicht illustriert.
- * Sie nutzt nicht (oder nur in geringem Maße) die Möglichkeiten von Hypertexten.

Diese Situation hat einige unerwünschte Konsequenzen. Zum einen wird es dem Einsteiger oder Umsteiger erschwert, sich systematisch Kenntnisse über Linux anzueignen. Dies schreckt viele potentielle Benutzer ab, die sich vielleicht gern mit diesem freien Betriebssystem beschäftigen würden.

Zum anderen mangelt es dem erfahrenen Benutzer - bis hin zum Administrator - an einem umfassenden Nachschlagewerk, das die schnelle Einarbeitung in neue Themen oder die Suche nach speziellen Informationen erleichtert.

Eine systematische Einarbeitung sowie ein gezieltes Auffinden von Informationen sind somit kaum möglich, das Lernen geschieht vielmehr "on-the-fly", unsystematisch und daher zeitintensiver als nötig.

4 Die Ziele von SelfLinux

SelfLinux möchte die in unseren Augen bestehenden Nachteile der [derzeitigen Linux-Dokumentation](#) schließen helfen. Im Vordergrund steht das Bestreben, einen Einblick in die Grundlagen der Arbeitsweise von Linux und der mit ihm verbundenen Werkzeuge zu geben. Der Benutzer und Administrator soll ein Verständnis für die Systemzusammenhänge entwickeln, das ihm bei der Problemlösung in der täglichen Arbeit behilflich sein wird. Von Anfang an möchten wir den Blick auch auf das richten, was "unter der Haube" geschieht, um Einblicke in die innere Logik des Systems zu geben, die vermutlich so viele Linux-Anhänger in ihren Bann zieht.

SelfLinux möchte vieles sein: Tutorial für Einsteiger, Ratgeber für Fortgeschrittene, bis hin zur Referenz für Profis. Aber natürlich können wir nicht alles auf einmal leisten. In den ersten Releases werden wir uns daher darauf konzentrieren, dem Einsteiger einen systematischen Weg zum fortgeschrittenen Linux-Wissen zu weisen. Wo bereits Texte verfügbar sind, werden wir jedoch auch den Fortgeschrittenen nicht vergessen. Ein

Nachschlagen nach bestimmten Themen lohnt allemal. Wir sind darüber hinaus sicher, dass auch diejenigen, die sich für fortgeschritten halten, viele elementare Themen haben brach liegen lassen, so dass sich auch für sie eine Lektüre der Einführungskapitel lohnen wird. Was die Verwendung als Nachschlagewerk angeht, so möge man noch etwas Geduld mit uns haben. Hunderte Autoren werden nötig sein, um den Umfang eines echten Nachschlagewerkes zu ermöglichen. Wir arbeiten daran - und Sie sind herzlich eingeladen, im Rahmen Ihrer Möglichkeiten (mit ihren Fachkenntnissen ...) daran teilzunehmen.

5 Das Konzept von SelfLinux

SelfLinux möchte einem Buch näher kommen als einer eher zufälligen Wissenssammlung. Es soll möglich sein, sich Schritt für Schritt durch das gesamte Tutorial oder auch nur in bestimmte Abschnitte einzuarbeiten. Gleichzeitig möchten wir jedoch nicht nur den Einsteiger in dieser Weise unterstützen. Auch der Fortgeschrittene will Zeit sparen und nicht alle Grundlagen zum wiederholten Male lesen.

Dies möchten wir realisieren, indem wir Texte einem bestimmten Kenntnisstand zuordnen, wir nennen es "Runlevelkonzept". Jeder Leser soll "nach Kenntnisstand" lesen können, d.h. genau das zu Gesicht bekommen, was er noch nicht kennt. Da wir für die Umsetzung des Runlevelkonzeptes eine große Textbasis benötigen, wird dieser Ansatz in den ersten Releases noch schwer zu realisieren sein, bleibt aber trotzdem das Ziel der Arbeit.

In den Diskussionen haben sich die folgenden drei wesentlichen Themen als "Eckpfeiler" für die Arbeit erwiesen:

- * Das Layout des Tutorials
- * Die Struktur der Kapitel
- * Die Arbeitsweise der Teams

5.1 Das Layout des Tutorials

Das Layout des Tutorials ist in seinen Grundzügen fertiggestellt. Dazu gehört die Menüstruktur ebenso wie Vereinbarungen über das Format von Überschriften, Absätzen und anderen wiederkehrenden Textstrukturen, die Festlegung von bestimmten Darstellungsarten für Kommandos, Dateiauszüge usw. Autoren verfassen ihre Texte im ASCII-Format und überlassen die Formatierung der Texte einem Layout-Team.

5.2 Die Struktur der Kapitel

Die Kapitel-Struktur von SelfLinux ist im Wesentlichen eine Baumstruktur, d.h. man navigiert vom Allgemeinen zum Speziellen und zurück, wie es beispielsweise vom Linux-Dateibaum her bekannt ist. Da es bei Hypertexten jedoch grundsätzlich keinen Anfang und kein Ende, kein oben und unten gibt, kann man durch Verlinkung auch andere Navigationswege ermöglichen. Daraus ergibt sich in unseren Augen ein riesiger Vorteil von SelfLinux gegenüber herkömmlicher Dokumentation: SelfLinux ermöglicht aufgrund seines Aufbaus den Einsatz in unterschiedlichen Ebenen der Wissensbeschaffung:

a) Verwendung als Tutorial

Es ist beispielsweise möglich, SelfLinux ganz linear als ein Tutorial zu verwenden, das beim Einfachen beginnt und zum Komplizierteren fortschreitet. Spätere Kapitel bauen also auf früheren auf. Die Benutzung entspricht der eines Buches.

b) Verwendung als Referenz

Für das schnelle Auffinden von Informationen ist meist eine Referenz das Mittel der Wahl. Sobald das vorliegende Textmaterial dies sinnvoll erscheinen lässt, ist eine Indizierung geplant, welche die Verwendung als Referenz ermöglichen soll. Neben einer Stichwortreferenz soll es auch eine Kommandoreferenz geben.

5.3 Die Arbeitsweise der Teams

Das SelfLinux-Team arbeitet virtuell, d.h. die Kommunikation findet praktisch ausschließlich per E-Mail statt,

entweder über die Mailingliste des Projektes oder unter einzelnen Teilnehmern zur Absprache von Details. Entscheidungen werden üblicherweise gemeinsam getroffen. Vorschläge aller Art sind willkommen, nach Möglichkeit mit Hinweisen zur Umsetzung.

Die Arbeit des SelfLinux-Teams gliedert sich in die folgenden Bereiche:

Autoren	Schreiben von eigenen Texten für SelfLinux im reinen Textformat. Die Themen orientieren sich an der Kapitelstruktur.
Scouts	Suchen von geeigneten Texten im Internet, Klärung des Copyrights, Einbindung in SelfLinux.
Layouter	Formatierung der Texte, Erstellung und Einbindung von Illustration.
Quality	Korrekturlesen, abschließende Überprüfung der Texte.
Maintenance	Technische Betreuung des CVS, der Homepage, der Mailingliste und ähnlichen Dingen.
XML	Betreuung, Verwaltung und Pflege von XML.
Translate	Übersetzung von Texten und Einbindung in SelfLinux.
Bugs	Suchen, finden, eintragen und fixen von Bugs.

Für jeden Bereich wurde ein eigenes Team mit jeweils einem Teamleiter gebildet.

6 Entstehung eines SelfLinux-Dokumentes

Aus den oben beschriebenen Anforderungen an SelfLinux ergibt sich folgender "Lebenslauf" eines Dokumentes für das Tutorial:

1. Ein Text wird erstellt

Ein Text wird entweder von einem Autor geschrieben, oder von einem Scout oder Übersetzer eingereicht. Dieser Text ist in reinem ASCII-Format geschrieben. Er enthält noch keinerlei endgültige Formatierungen etc.

2. Erste Qualitätskontrolle

Hier überprüft die Quality, ob der Text den Richtlinien für "Plain"-Texte in SelfLinux entspricht, ob der Text fehlerfrei, ob er logisch aufgebaut ist, ob der Autor angegeben ist u.v.a.m.

3. Der Text wird an das Layout-Team übergeben

Nun wird vom Layout das Textdokument mit den für SelfLinux spezifizierten XML-Tags versehen, etwaige Bilder werden eingefügt, das Format festgelegt etc.

4. Zweite Qualitätskontrolle

Danach überprüft die QA (das Quality-Team) abermals, ob das Dokument den festgelegten Richtlinien entspricht.

5. Der Text wird für ein Release freigegeben

Es ist nun ein fertiges Dokument entstanden, das für eine Veröffentlichung im Rahmen von SelfLinux vorgemerkt wird. Solche Releases werden dann von Zeit zu Zeit auf der offiziellen Homepage von SelfLinux eingearbeitet und zum Download angeboten.

Während des gesamten Arbeitsablaufs ist die Kommunikation zwischen den beteiligten Teams sowie die Dokumentation der Schritte von entscheidender Bedeutung. Zur Kommunikation dient hauptsächlich die Mailingliste, aber auch E-Mail zwischen den Projektleitern oder innerhalb der Teams. Zur Dokumentation und Versionskontrolle setzen wir CVS ein. So hat jeder Entwickler und jedes Team die Möglichkeit, mit anderen zusammen an ein und demselben Dokument zu arbeiten, Zwischenstände zu sichern, die geleisteten Arbeitsschritte zu kennzeichnen und vieles mehr.

7 Ausblick

Wir denken, dass wir mit der Konzeptionierung von SelfLinux, den Ideen, der Projektverwaltung und den gebildeten Teams auf einem guten Weg sind.

Im Juli 2002 veröffentlichten wir mit dem Basisrelease einen ersten Grundstein. Seither bauen wir mit regelmäßig erscheinenden Versionen darauf auf und erweitern so den Umfang von SelfLinux.

Mit SelfLinux 0.4.0 haben wir die Nummerierung unserer Release dem GNU-Stil angepasst. Mit geraden Nummern werden stabile Versionen gekennzeichnet, mit ungeraden unsere Testversionen. Dadurch hoffen wir, etwaige Verständnisprobleme, wie sie das Basisrelease mancherorts ausgelöst hat, in Zukunft zu verhindern.

Mit den bisherigen Versionen sind wir aber erst einen kleinen Teil des Weges gegangen. Bis zu einer vollständigen und nützlichen Dokumentation, die wir uns vorstellen, ist es noch sehr weit.

Deshalb suchen wir für alle Teams noch Mitwirkende, die ein wenig Zeit mitbringen und ihre Energie in ein wichtiges und gutes Projekt, investieren wollen: Die Entwicklung einer freien, funktionalen, nutzer- und bedienfreundlichen, deutschsprachigen Linux-Dokumentation.