SelfLinux-0.10.0

Was ist eine Shell?

Autor: Matthias Kleine (*kleine_matthias@gmx.de*)
Formatierung: Matthias Hagedorn (*matthias.hagedorn@selflinux.org*)

Inhaltsverzeichnis

1 Einleitung

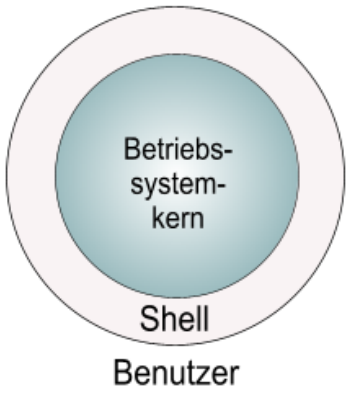
Das vorliegende Kapitel gibt bewusst keine Einführung in technische Details. Es ist insbesondere an Benutzer gerichtet, die noch keine große Erfahrung im Umgang mit Shells haben und die wissen möchten, warum Shells überhaupt verwendet werden und welchen Nutzen sie haben. Wer nach technischen Einzelheiten zum Umgang mit Shells sucht, sei auf die beiden Kapitel Einführung in die Bourne Again Shell und Shellprogrammierung verwiesen.

In diesem Kapitel werden insbesondere die folgenden Fragen beantwortet:

* Welche Rolle spielt eine Shell innerhalb des Gesamtsystems?
* Wodurch erleichtert eine Shell die tägliche Arbeit?
* Wie trägt die Shell zur Einrichtung einer persönlichen Umgebung bei?
* Gibt es nur eine oder gibt es unterschiedliche Shells?

2 Ein Mittler zwischen Benutzer und Betriebssystem-Kern

Weil die direkte Kommunikation mit dem Betriebssystem-Kern für einen Benutzer viel zu komplex wäre, ist eine vereinfachte Benutzer-Schnittstelle erforderlich. Neben einer grafischen Schnittstelle wie dem X Window System wird diese Leistung vor allem von einer Shell bereitgestellt. Der englische Ausdruck **Shell**, zu Deutsch etwa **Schale** oder **Ummantelung**, drückt diesen Sachverhalt bereits aus. Die Übersetzung oder Symbolisierung als **Muschel** hat dabei wohl mehr mit Spieltrieb und Anschaulichkeit als mit einem echten technischen Hintergrund zu tun. Jedenfalls lässt sich eine Shell als eine Schicht zwischen Betriebssystem und Benutzer verstehen. Wir veranschaulichen dies in der folgenden Grafik:



Funktionsweise einer Shell

Während Benutzer, die noch nicht häufig mit Shells in Berührung gekommen sind, den wartenden Eingabeprompt einer Shell als trist und abweisend, ja sogar als Hindernis empfinden mögen, wurden Shells doch mit der gegenteiligen Absicht entwickelt: Sie sollten die tägliche Arbeit vereinfachen und erleichtern.

3 Die Interpretation der Kommandozeile

Die Hauptaufgabe einer Shell besteht darin, Kommandos entgegenzunehmen und das Betriebssystem um ihre

Ausführung zu bitten. Das klingt einfach, stößt aber in der Praxis auf besondere Probleme und Anforderungen. Eine Shell hat daher zunächst eine Interpretation der Kommandozeile zu leisten. Sie interpretiert einzelne Zeichen oder Worte der Eingabe und ersetzt sie ggf. durch neue Zeichen oder Worte. So stehen beispielsweise Variablen für bestimmte Zeichenketten und werden zunächst von der Shell durch ihren Inhalt ersetzt, bevor der Aufruf eines Kommando erfolgt. Das gleiche gilt für Platzhalterzeichen wie * oder ?, die für die Angabe von Zeichenmustern verwendet werden können. Diese und zahlreiche weitere Mechanismen kann der Benutzer verwenden, um seine Kommandozeilen knapper und effektiver zu formulieren.

4 Die Kombination von Kommandos

Häufig ist es auch nicht nur ein einzelnes Kommando, das ausgeführt werden soll. Kommandos können beispielsweise so miteinander verbunden werden, dass die Ausgabe eines Kommandos zur Eingabe eines weiteren Kommandos wird. Auch wiederholte oder bedingte Ausführungen in Form von Schleifen und **wenn - dann**-Konstrukten sind wünschenswert und werden daher von vielen Shells ermöglicht. Und schließlich möchte man häufig eine bestimmte wiederkehrende Folge von Kommandos ausführen lassen, die man daher in einer Datei niederschreibt und für die spätere Ausführung abspeichert. Die Konfiguration eines Linux-Systems erfolgt häufig über solche Dateien, die als Shell-Skripte bezeichnet werden.

5 Bereitstellung einer persönlichen Arbeitsumgebung

Bei der täglichen Arbeit ist des weiteren die Konfiguration einer typischen Arbeitsumgebung von Bedeutung. So möchte ein Benutzer beispielsweise meist mit einer passenden Spracheinstellung arbeiten, seinen Lieblingseditor zum Editieren von Texten verwenden, verwendeten Programmen Information über seine Vorlieben mitteilen, die Form seines Eingabepromptes ändern und vieles andere. All dies lässt sich über sogenannte Shellvariablen konfigurieren, auf Wunsch auch dauerhaft in einer Konfigurationsdatei, damit die Einstellungen nicht bei jeder Anmeldung wiederholt werden müssen. Eine Shell bietet ihrem Benutzer somit eine konfigurierbare, persönliche Umgebung, um typische Arbeitsabläufe möglichst komfortabel und fehlerfrei abwickeln zu können.

6 Unterschiedliche Shells

Im Laufe der Zeit wurden unterschiedliche Shells entwickelt. Je nach Erfahrungshintergrund bevorzugen Anwender einzelne Shells, die es ihnen besonders gut ermöglichen, ihre jeweiligen Ziele zu erreichen. Die oben beschriebenen Anforderungen werden zwar von jeder Shell erfüllt, gelegentlich können jedoch besondere Anforderungen wie beispielsweise besonders komfortable Programmierbarkeit, Unterstützung spezieller Programmier-Konstrukte oder Kompatibilitätserwägungen eine Rolle bei der Auswahl der Shell spielen.

Die Mutter aller Shells stellt gewissermaßen die **Bourne Shell** dar. Unter Unix darf man sicher sein, zumindest eine mit der Bourne Shell kompatible Shell anzutreffen, und die meisten Skripte sind auch heute noch in der Bourne Shell-Syntax gehalten. In der Linux-Welt ist zweifellos die **Bourne Again Shell** (**bash**) am weitesten verbreitet, die zur Bourne Shell aufwärtskompatibel ist und darüber hinaus zahlreiche Erweiterungen erfahren hat. Aber auch andere Shells wie etwa die **C-Shell** (**csh**) und ihre Nachfolger sowie **ash** oder **zsh** sind in der Praxis immer wieder anzutreffen. Wenn keine besonderen Anforderungen eine Rolle spielen, ist die bash für den Einstieg eine gute Wahl.

7 Zusammenfassung

Wir haben in diesem Kapitel keine technischen Details besprochen, sondern uns der Frage gewidmet, warum

Shells überhaupt verwendet werden und welche Leistung sie für die Arbeit mit einem Computer erbringen. Der Urgrund für die Entwicklung von Shells war die Bereitstellung einer Kommunikations-Schicht zwischen Betriebssystem und Benutzer, damit dieser nicht die komplexen Eigenheiten des Betriebssystemes selbst kennenlernen musste. In der täglichen Arbeit ergab sich dann der Wunsch nach zusätzlichen Mechanismen, welche dem Benutzer halfen, typische Arbeitsschritte effizienter zu bewältigen. Die Entwicklung unterschiedlicher Shells ist in diesem Sinne nur ein Ausdruck unterschiedlicher Anforderungen an den Komfort und die Leistungsfähigkeit der Shell. Man sollte die verschiedenen Mechanismen ebenso wie die Vielfalt der Shells weniger als eine Hürde, sondern als einen Luxus verstehen.

Wenn Sie diesen Luxus näher kennenlernen wollen, sei daher hier nochmals auf die beiden Kapitel Einführung in die Bourne Again Shell und Shellprogrammierung verwiesen, welche sich dann auch den technischen Einzelheiten mit größerer Ausführlichkeit widmen.