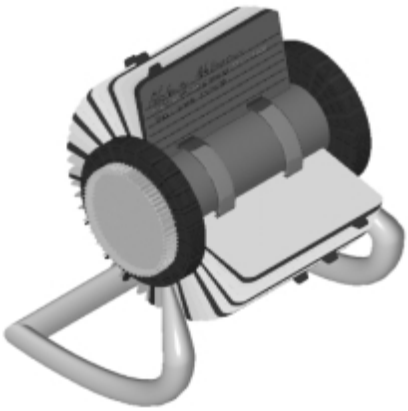
Registry für Linux



PAUL WEINSTABL

Microsoft hat für Windows die Registry erfunden, die einer der Hauptgründe ist, warum ältere Installationen des Betriebssystems langsam und instabil werden. Warum soll man deshalb eine Registry oder ähnliches bei Linux und anderen unix-artige Systeme einführen? Bietet diese Technik tatsächlich Vorteile gegenüber Konfigurationsdateien? Dieser kontroverse Beitrag geht diesen Fragen nach.

Seit einiger Zeit gibt es ein Projekt, das sich zum Ziel gesetzt hat, ein allgemein verwendbares und zentral verfügbares Konfigurationssystem für Linux und andere unixartige Betriebssysteme zu schaffen. Durch dieses System soll sichergestellt werden, daß die Konfigurationsdaten der einzelnen Programme in einem zentralen Schema mit sogenannten Schlüssel-Wert-Paaren zusammengefaßt werden. Das Projekt heißt Elektra und ist auf Sourceforge gehostet.

Bezug der Software und Installation

Die Idee zu Elektra entstand aus der Frustration darüber, daß jedes Programm seine Konfigurationsinformationen in einer eigenen Datei und in einem eigenen Datenformat ablegt. Dadurch wird es schwierig, etwa Systemupdates automatisch durchzuführen. Auch die Integration mehrerer Softwaresysteme wird dadurch unnötig erschwert. Avi Alkalay, der Initiator von Elektra, bemerkt dazu weiter, daß ein Softwareentwickler unnötig viel Zeit dazu verwenden muß, Code zum Parsen von Konfigurationsdateien zu schreiben, nur um seine Software mit anderer Software zu integrieren. Ebenso können verschiedene Linux-Distributionen die Konfiguration von Software in eigenen Formaten an verschiedenen Orten ablegen. Und das alles muß der Systemadministrator berücksichtigen und vor allem auch wissen. Auch ist es im momentanen Zustand nur sehr schwer möglich, herauszufinden, welche Änderungen in einer bestimmten Konfigurationsdatei durchgeführt wurden.

Da es sich bei Elektra primär um eine Bibliothek zum Referenzieren und Bearbeiten von Schlüssel-Wert-Paaren handelt, gibt es nicht viel zu installieren. Die Software kann von der Webseite *http://elektra.sourceforge.net/* bezogen werden. Die Binärpakete von Elektra installieren mehrere Bibliotheken im Bibliotheksverzeichnis */usr/lib*. Außerdem wird ein Kommandozeilenwerkzeug zum Bearbeiten der Schlüsseldatenbank mit dem Namen *kdb* in */usr/bin* eingefügt, zusätzlich ein Shellskript *elektraenv.sh* in */etc/profile.d*, damit nun Elektra die Einstellung der Umgebungsvariablen vornehmen kann.

Bild 1 zeigt, daß das Skript */etc/profile.d/elektraenv.sh* vor dem Öffnen des Terminalfensters abgearbeitet wurde. Es ist auch ein lehrreiches Beispiel für die Entwicklung eigener Skripten, um selbsterstellte Software und ihre Konfigurationsparameter in Elektra zu integrieren.

Die Struktur der Namensräume

Zusätzlich zu den oben erwähnten Dateien werden noch mehrere Shellskripte im Verzeichnis */usr/share/doc/elektra* installiert, die verschiedene Konfigurationsdateien im Verzeichnis */etc* Schlüssel-Wert-Paaren umzuwandeln können. So liefert etwa das Skript *users-convert* Schlüssel-Wert-Paare für die Dateien */etc/passwd*, */etc/group* und */etc/shadow*. Natürlich

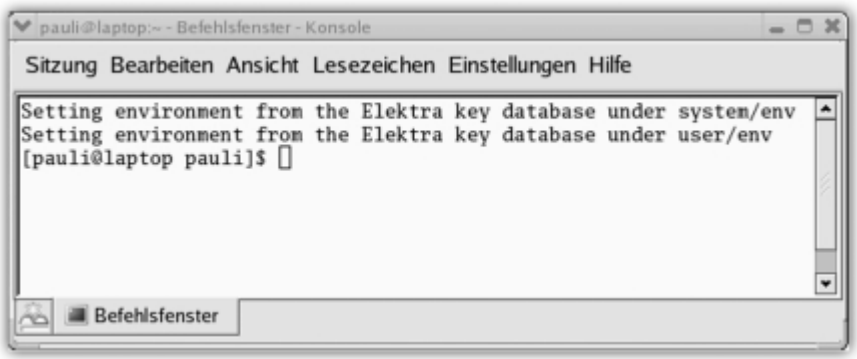


Bild 1: Ein geöffnetes Terminalfenster

Datentyp	Beschreibung
Verzeichnis	Es kann selbst keine Werte enthalten, dient als Gruppierung für nachfolgende Schlüssel-Wert-Paare. Im Dateisystem wird dieser Datentyp ebenfalls durch ein Verzeichnis repräsentiert.
String	Wird als Text behandelt und gibt einen Schlüsselnamen an.
Link	Link zu einem anderen Schlüssel. Wird im Dateisystem ebenfalls als symbolischer Link abgebildet.
Binary	Ein Stream von Bytes, nicht notwendigerweise Text. Dieser Datentyp sollte möglichst vermieden werden.

Tabelle 1: Die Schlüsselinformationen der in Elektra verfügbaren Datentypen.

muß man für den Ablauf dieses Skripts die entsprechende Berechtigung als Systemverwalter besitzen.
So gibt es im oben erwähnten Verzeichnis Skripten zur Konvertierung der Netzwerk-Parameter, der Inhalte in */etc/hosts*, */etc/fstab*, */etc/inittab*, zur Konvertierung der Hardwarekonfiguration und zur Konvertierung der XFree-Konfiguration.
Elektra sieht grundsätzlich – wohl angelehnt an die Namen bei Windows – zwei unterschiedliche Namensräume für das Gesamtsystem vor:

- Die Domäne *system*: Im ersten Namensraum */system* werden alle globalen Applikationsschlüssel und deren Konfiguration abgelegt. Auch andere systemweite Parameter, wie etwa die Hardwarekonfiguration der verwendeten Maschine, können hier abgelegt werden. Dieser Namensraum entspricht den Dateien unter */etc*.
- Die Domäne *user*: Im zweiten Namensraum */user* werden für alle Systembenutzer die entsprechenden Konfigurationsdaten abgelegt. Weitestgehend entspricht dieser Namensraum den versteckten Dateien (mit vorangestelltem Punkt) im jeweiligen Heimatverzeichnis des Benutzers. Im Gegensatz zum System-Namensraum ist der Benutzernamensraum dynamisch, das heißt, daß einzelne Werte in verschiedenen Sub-Namensräumen unterschiedliche Ausprägungen annehmen können.

Physikalische Ablage

Physikalisch werden die Daten in einzelnen Dateien abgelegt, und zwar so, daß es für jeweils einen Namen eine eigene Datei gibt. Es handelt sich bei diesen Dateien um normale Textdateien, die UTF-8-codiert sind. Sie müssen nicht mit dem vorgesehenen Werkzeug *kdb* bearbeitet werden, sondern können in jeden Texteditor geladen werden.
Außerdem werden die normalen Berechtigungskonzepte des Linux-Dateisystems berücksichtigt. Diese im Elektra-Jargon als »Metadaten« bezeichneten Informationen können mit Hilfe von Elektra gezielt manipuliert und editiert werden.
Die Dateien für den Namensraum *system* werden physikalisch unter dem Verzeichnis */etc/kdb* abgelegt (zum Beispiel */etc/kdb/system/sw/regedit/gui/iconDir*). Die Dateien für die Benutzernamensräume werden im *$HOME*-Verzeichnis des jeweiligen Benutzers unter dem Verzeichnis *.kdb* abgelegt (beispielsweise also unter */home/pauli/.kdb/user/sw/regedit/gui/height*).
Daß das System der Namensräume auf dem Dateisystem basiert, zeigen schon die für einen neuen Schlüssel verfügbaren Attribute. So kann man ein weiteres Verzeichnis (Directory) anlegen, um dort irgendwelche Wertpaare zu speichern (String oder Binary), wie zum Beispiel ein Verzeichnis für die Konfiguration von *john*, oder aber es werden Werte unter einem bestimmten Schlüssel (= Namen) abgelegt. Außerdem kann noch ein Link angelegt werden. Dabei handelt es sich tatsächlich um einen Link auf ein anderes Verzeichnis oder eine andere Datei; er funktioniert bisher aber nicht über Dateisystemgrenzen hinweg.
Die Schlüssel dienen als Basis für den Zugang zu den Konfigurationsdaten. Wie schon beschrieben, werden die Informationen in einfachen Textdateien abgelegt. Dafür stehen die in Tabelle 1 aufgelisteten Datentypen zur Verfügung.
Zu den Schlüsseln sind weiter noch Metadaten verfügbar. Sie stellen die für die physikalische Abbildung der Daten im Dateisystem notwendigen Parameter dar: Benutzername und

Kommandozeile	Funktion
set -t dir user/pauli	Anlegen eines neuer Benutzer-Schlüssels.
rm user/heini	Löschen des Benutzers »heini«.
set -t dir user/pauli/sw	Ein neuer Schlüssel-Eintrag für »pauli«.
set user/pauli/sw/MyApp - "Textverarbeitung"	Setzen eines Schlüssels mit Wert.
ls -lR user	Anzeige der Struktur ab dem Schlüssel »user« und zwar rekursiv durch alle Unterverzeichnisse.

Tabelle 2: Einige Kommandozeilen von kbd und was bei ihrem Aufruf geschieht

```
[root@laptop pauli]# kdb ls -laR user
drwxr-xr-x   root  root  4096 Dec 23 15:44 user/heini
drwxr-xr-x   root  root  4096 Dec 23 15:46 user/heini/sw
-rw-r--r--   root  root    55 Dec 23 15:47 user/heini/sw/First_App
drwxr-xr-x   root  root  4096 Dec 22 15:45 user/john
drwxr-xr-x   root  root  4096 Dec 22 15:45 user/john/mailagent
-rw-r--r--   root  root    19 Dec 22 15:45 user/john/mailagent/mail
drwxr-xr-x   root  root  4096 Dec 22 15:40 user/pauli
drwxr-xr-x   root  root  4096 Dec 22 15:40 user/root
drwxr-xr-x   root  root  4096 Dec 22 15:37 user/sw
drwxr-xr-x   root  root  4096 Dec 22 15:37 user/sw/regedit
drwxr-xr-x   root  root  4096 Dec 22 15:37 user/sw/regedit/gui
-rw-r--r--   root  root    73 Dec 22 15:50 user/sw/regedit/gui/height
-rw-r--r--   root  root    72 Dec 22 15:50 user/sw/regedit/gui/width
-rw-r--r--   root  root    76 Dec 22 15:50 user/sw/regedit/gui/x
-rw-r--r--   root  root    76 Dec 22 15:50 user/sw/regedit/gui/y
[root@laptop pauli]#
```

Bild 2: Einträge im Dateisystem

Benutzergruppe des Dateibesitzers, Zugriffsberechtigungen, Zeit des letzten Zugriffs und der letzten Modifizierung und Kommentare für den Schlüssel.

Mit diesen Informationen kann man jetzt selbst Schlüssel und Werte mit Hilfe des Kommandozeilenwerkzeugs *kdb* anlegen und manipulieren. Einige Beispiele listet Tabelle 2 auf.

Man sollte immer beachten, daß alle getätigten Einträge durch entsprechende Verzeichnisse beziehungsweise Dateien physikalisch im Dateisystem repräsentiert werden. Deshalb muß man nicht zwingend das Werkzeug *kdb* verwenden, um ähnliche oder dieselben Resultate zu erzielen. Die Systemprogramme *mkdir*, *touch* und so weiter erfüllen den gleichen Zweck.

Schnittstelle für eigene Programme

Bild 2 zeigt ein weiteres Beispiel mit derartigen Einträgen im Dateisystem. Auf der Website von Elektra wird ein grafisches Werkzeug zur Manipulation der Schlüssel-Wert-Paare erwähnt, nämlich *regedit*. Es ist von der Webseite *http://www.livejournal.com/users/gregorburger/* erhältlich und stellt in etwa die gleichen Funktionen wie das Kommandozeilentool *kdb* in einer grafischen Umgebung zur Verfügung.

Leider gibt es damit keine Möglicheit, als Systemverwalter in der grafischen Oberfläche uneingeschränkt die Registry zu bearbeiten und zu verwalten, weshalb es hier nur erwähnt werden soll. Man sollt statt dessen lieber zum *kdb* greifen.

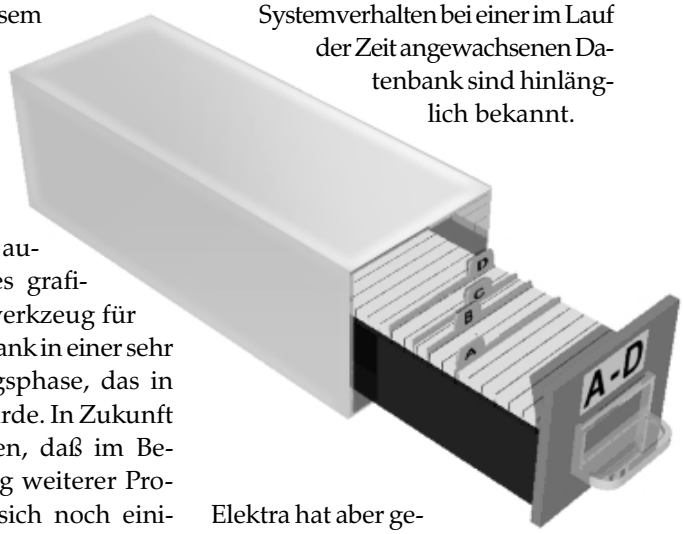
Eines der Ziele von Elektra ist es, Softwareentwicklern zu erleichtern, Parameterinformationen zu lesen, in eigenen Programmen zu verwenden und auch entsprechende Konfigurationsoptionen für Eigenentwicklungen geordnet nach den Vorgaben von Elektra ins jeweilige System einzutragen, damit anderen Programmen danach ebenfalls diese Optionen zur Verfügung stehen und diese die eingefügten Daten leicht auslesen können. Diesem Zweck dient das API von Elektra.

Es gibt Bindings für Programmiersprachen wie zum Beispiel C++, Java, Python und Ruby, außerdem ein weiteres grafisches Verwaltungswerkzeug für die Regis-try-Datenbank in einer sehr frühen Entwicklungsphase, das in Java geschrieben wurde. In Zukunft ist damit zu rechnen, daß im Bereich der Anbindung weiterer Programmiersprachen sich noch einiges tun wird.

Fazit

Die Idee, eine Registry für Linux zu erstellen, ist meines Erachtens nach interessant. Auch das Konzept einer besseren Integration der verschiedensten Softwareumgebungen und einer einheitlichen Schnittstelle für die Rechnerkonfiguration muß begrüßt werden. Insbesondere Umsteiger von Windows werden sich damit leicht tun, denn daß in von Unix-Systemen jedes Programm seine Konfigurationsinformationen in einer eigenen Datei und in einem eigenen Datenformat ablegt, ist gerade für Einsteiger schwer zu durchschauen und gewöhnungsbedürftig. Problematisch könnte auf großen Systemen mit vielen verschiedenen Softwareinstallationen die physikalische Speicherung der Konfigurationsoptionen in einzelnen Verzeichnissen und Dateien sein, denn leicht könnten dem Dateisystem dabei die verfügbaren i-Nodes ausgehen. Vielleicht sollte in diesem Bereich noch über eine alternative Möglichkeit der Ablage dieser Informationen nachgedacht werden.

Allgemein erfolgt der Zugriff auf Schlüsselinformationen und Werte im Test sehr rasch, da die benutzten Dateien grundsätzlich sehr klein sind. Wie es in der Realität bei ausgiebiger Nutzung aussehen wird ist, steht auf einem anderen Blatt. Die Schwächen der Windows-Registry wie Unüberschaubarkeit und mangelhaftes Systemverhalten bei einer im Lauf der Zeit angewachsenen Datenbank sind hinlänglich bekannt.



Elektra hat aber gegenüber ähnlichen Ansätzen den Vorteil, daß es sich dabei um eine Bibliothek mit äußerst wenig Abhängigkeiten handelt, die schon während des Bootens verwendet werden kann. Damit könnte Elektra bei dem, der dies mag, tatsächlich zu so etwas wie einer allgemein verwendeten Registry für Linux und ähnliche Betriebssystemumgebungen werden. Jedenfalls ist das Projekt interessant genug, um auch weiterhin genau beobachtet zu werden. ◆