Internet-Suchmaschinen auf der Kommandozeile befragen

Prompt gesucht

Tilman Wittenhorst

Das Internet lässt sich auf viele Arten durchsuchen. Die Kommandozeile verspricht passende Ergebnisse mit weniger Eingaben, wenn man einige Tricks beachtet.

Wer im Internet sucht, der findet – meistens zu viel. Grenzt man die Suche durch Bedingungen ein, lichtet sich zwar das Dickicht der Ergebnisse, jedoch geraten die Eingaben immer länger. Sucht man beispielsweise nach PDF-Dateien, fügt man den Suchbegriffen *ext:pdf* hinzu. Sollen die Ergebnisse von einer bestimmten Webseite stammen, ergänzt man etwa *site:de.wikipedia.org*. Die bei Google „erweiterte Suche" genannten Einstellungen sind zahlreich. Wer sie häufig benutzt, kann sich durch Tricks auf der Kommandozeile Zeit und Mühe sparen.

Ruft man den Browser per Tastatur mit der Such-URL von Google auf und ergänzt die Suchbegriffe, muss jedes Leerzeichen als *%20* kodiert werden. Das lässt sich in einer Shell-Funktion automatisieren, der man die Suchbegriffe als Parameter übergibt. Die im Linux-Umfeld verbreitete Bourne-again-Shell, kurz Bash, liest bei jedem Start die Datei *.bashrc* ein. Mit einem Editor ergänzt man sie um selbst geschriebene Funktionen. Die folgende namens *blank()* kodiert Leerzeichen und dient als Grundlage für weitere Suchfunktionen:

```
blank() {
  echo $@ | sed 's/[[:blank:]]/%20/g'; }
```

Die Variable $@ enthält alle Suchbegriffe in einer Zeichenkette. Das *echo*-Kommando reicht sie durch die Pipe (senkrechter Strich) an den Stream-Editor *sed* weiter. Dieser ersetzt mit *s* („substitute") das Leerzeichen *[[:blank:]]* durch die Zeichenfolge *%20*. Das *g* für „global" erfasst alle Leerzeichen in der Zeichenkette, nicht nur das erste.

Kodieren ist Pflicht

Eine zweite Funktion, *gg*, durchsucht die deutsche Google-Webseite und greift auf *blank* zurück:

```
gg() {
 firefox https://www.google.de/search? ⤷
                                q=`blank $@`; }
```

Die Gravis (`) weisen Bash an, die Ausgabe von *blank* an dieser Stelle einzusetzen. Änderungen an der Datei *.bashrc* wirken sich nur auf neue Shell-Sitzungen aus. In geöffneten Sitzungen muss man sie mit dem Kommando *source .bashrc* aktivieren. Anschließend öffnet *gg bourne-again shell* den Browser und sucht bei Google nach den eingegebenen Begriffen.

Weitere Sucheinstellungen ergänzt man mit einem Pluszeichen. Wer beispielsweise bei Intel nach technischen PDF-Datenblättern (Suchbegriff „spec") sucht, definiert eine neue Funktion:

```
intel() {
 firefox https://www.google.de/search? ⤷
   q=`blank $@`%20spec+site:intel.com+ext:pdf; }
```

Ein knappes *intel e5-2699* führt zum Ziel. Jedoch bleibt die Notwendigkeit, das gewünschte Ergebnis anzuklicken. Die Suche auf einer bestimmten Webseite statt in der Suchmaschine kann das vermeiden, etwa bei Wikipedia:

```
wp() {
 firefox https://de.wikipedia.org/w/index.php ⤷
                                ?search=`blank $@`; }
```

wp bourne-again shell öffnet den Wikipedia-Artikel – allerdings nur, wenn die Suchbegriffe exakt sind. Viele Webseiten besitzen eine eigene Suche. Wie sich deren URL jeweils zusammensetzt, muss durch einen Test ermittelt werden.

Bash-Funktionen lassen sich unter Linux und etlichen Unix-Varianten sowie Mac OS X einsetzen. In letzterem ändert sich lediglich das Starten des Browsers: Statt *firefox* muss es *open –a Firefox.app* heißen. Wer mit Windows arbeitet, kann Bash mit der Emulationsumgebung Cygwin nachrüsten. Sie erlaubt den Betrieb von Unix-Software unter Windows.

Die Webseite *goosh.org* bietet eine Shell in JavaScript zum Befragen von Suchmaschinen. Bestätigt man die eingegebenen Suchbegriffe mit ENTER, zeigt goosh die ersten vier Ergebnisse als Text an. Die Zahlen Eins bis Vier gefolgt von ENTER öffnen die entsprechende URL in einem neuen Reiter. ENTER ohne weitere Eingabe zeigt die nächsten Ergebnisse.

Goosh verarbeitet Googles erweiterte Suchparameter wie *site:* und *ext:*. Es kennt Kürzel für andere Suchmaschinen, die der Befehl *help* dokumentiert. *wiki* schaut bei Wikipedia nach, *blogs* nutzt die Google-Blogsuche und *lucky* öffnet das erste Suchergebnis.

Elvis lebt in /usr/lib

Die Unix-Software *surfraw* sucht auf beliebigen Seiten und nutzt deren jeweils unterschiedliche Optionen. Das bewerkstelligen „elvis" genannte umfangreiche Shell-Funktionen in */usr/lib/surfraw/*, die das Werkzeug nach Bedarf erweitern. Ein *surfraw –elvi* führt alle auf. Derzeit gibt es mehr als hundert, darunter Bugzilla, Google-Cache, die MySQL-Dokumentation, SourceForge und Wolfram Alpha.

Welche Optionen ein elvi mitbringt, erklärt *surfraw [ELVINAME] –h*. Beispielsweise lässt sich beim Wikipedia-Elvi die Sprache mit *–l=en* auf Englisch festlegen:

```
surfraw wikipedia -l=en bourne-again shell
```

Die Software besitzt zudem globale Optionen wie *–p*, das die Such-URL ausgibt und den Browser umgeht. Durchsucht beispielsweise ein Skript eine Logdatei nach Fehlercodes, kann es deren Such-URLs bei Bugzilla in eine E-Mail an den Administrator schreiben. Internetsuche an der Kommandozeile soll schließlich nicht nur Tastenanschläge sparen, sondern darf auch der Bequemlichkeit dienen. (tiw)

Alle Links: www.ix.de/ix1410131

© Copyright by Heise Zeitschriften Verlag
Persönliches PDF für alle Leser der iX aus 30625 Hannover