Volltextsuche mit Swish-E

Finden im Sauseschritt

Eine Website mit einer Suchfunktion versehen ist Stand der Technik. Um aber die eigene Festplatte umfassend und mit all den Dateiformaten abseits von HTML zu durchsuchen, braucht es etwas Effizienteres als Grep oder einen einfachen Index: Swish-E. Rolf Strathewerd

Der Speicherplatz heutiger Festplatten ist zwar groß, dennoch gelingt es immer wieder, sie mit allerlei Dateien vollzustopfen. Selbst die ordentlichste Verzeichnisstruktur hilft wenig, wenn man einen bestimmten Text, ein altes Skript oder ein spezielles Bild wiederfinden will. Mangels intelligenter Computer bleibt heute nur der Griff zur Volltextsuche. Zu den besten Ergebnissen führt das Simple Web Indexing System for Humans Enhanced, kurz Swish-E [1] genannt. To swish heißt so viel wie sausen – das passt im doppelten Sinne. Die Indizierung läuft recht flott und auch die Suche liefert ihre Ergebnisse ohne merkliche Verzögerung.

Swish-E wurde 1995 unter dem Namen Swish von Kevin Hughes als Webindizierer für ein kommerzielles Umfeld geschrieben. Zwei Jahre später fragte ihn die Universität Berkeley, ob sie Fehler im Code beheben und das Projekt weiterentwickeln könne. Daraus entstand das heutige Swish-E 2, das von einem kleinen Team gepflegt wird und mittlerweile den Rahmen eines einfachen Webindizierers sprengt.

Für Swish-E besteht die Welt nur aus Text, HTML und XML. Wird er ohne Optionen gestartet, interpretiert der Indizierer alle Dateien, auf die er stößt, als eines dieser Formate und schreibt einen Default-Index. Dazu startet das Programm im aktuellen Verzeichnis und arbeitet sich zu den darunter liegenden Verzeichnissen vor.

Index erstellen

Das ist nur selten das gewünschte Verhalten. Eine passende Konfigurationsdatei (Listing 1) bringt das Tool auf den rechten Pfad: Die Datei legt fest, welches Verzeichnis abzuarbeiten ist (»IndexDir«), wie die Indexdatei heißt (»IndexFile«) und auf welche Sorten von Dateien sich die Indizierung beschränken soll (»IndexOnly«). Das Kommando »swish-e -c localweb.conf« startet daraufhin die Indizierung. Gerade beim Testen ist es sinnvoll, die Kommandozeile noch um die Option »-v3« zu erweitern. Mit ihrer Hilfe zeigt Swish-E ausführlich, was es gerade analysiert.

Um im erzeugten Index zu suchen, genügt es, Swish-E mit etwas anderen Parametern aufzurufen. So fahndet etwa »swish-e -f localweb.index -w "Hallo"« im eben erstellen Index »localweb.index« nach dem Wort »Hallo«. Für komplexere Suchen stehen »*« als Wildcard sowie die logischen Operatoren »and«, »or« und »not« zur Verfügung.

Da die Welt nicht nur aus Text, HTML und XML besteht, kann Swish-E mit Filtern auch andere Formate verarbeiten. Diese Filter verwandeln den neuen Dateityp in eines der drei bekannten Formate. Das funktioniert zum Beispiel mit Jpeg und Open Office, aber auch mit Mailboxen und Datenbanken. Meist ist dafür nicht viel mehr erforderlich, als den Namen des Konvertierungsprogramms in die Konfigurationsdatei einzutragen, nur selten muss man ein kleines Shellskript verfassen.

Swish-E

Name: Swish-E, Simple Web Indexing System for Humans Enhanced
Lizenz: GPL und LGPL
Status: Stabil (Version 2.4.1)
Kategorie: Indizierer und Suchmaschine, geeignet für lokale Suche (Festplatte, CD etc.) und für Websites
Plattformen: Swish-E ist in C programmiert und wurde auf vielen Plattformen getestet, unter anderem Linux, Sun Solaris, Dec Alpha, BSD, Linux, Mac OS X und Open VMS. Es enthält auch Perl-Module.
Homepage: [http://www.swish-e.org]

Wie für die meisten verbreiteten Formate findet sich auch ein Programm, das PDF in schlichten Text umwandelt: »pdftotext« [2]. Damit Swish-E es benutzt, muss der Konverter in der Konfigurationsdatei eingetragen sein.

PDF-Dokumente indizieren

Das Schlüsselwort »IndexContents« (Listing 2) teilt dem Indizierer mit, dass er Dateien mit dem Suffix ».pdf« als Textdateien anzusehen hat. Da sie das nicht wirklich sind, verknüpft »FileFilter« das Suffix ».pdf« mit dem Programm »pdftotext«. Trifft Swish-E bei der Indizierung auf eine PDF-Datei, schickt es sie durch den Filter, der den entstandenen Text auf die Standardausgabe schreibt. Dort holt ihn Swish-E ab und verarbeitet ihn. Alle Filter für Fremdformate müssen das Ergebnis ihrer Umwandlungsarbeit an Stdout ausgeben.

Swish-E verfügt für die Textanalyse zwar über einen internen Parser, kann aber auch mit der »libxml2« arbeiten. Mit der Ergänzung »2« hinter den Formatangaben »TXT«, »HTML« oder »XML« benutzt der Indizierer immer Libxml2, bei einem Sternchen bleibt die Verwendung dieser Library optional und ohne jedes Anhängsel kommt immer der interne Parser zum Einsatz. In Listing 2 weist das Sternchen hinter »TXT« (Zeile 2) Swish-E dazu an, mit der Libxml2 zu arbeiten, falls sie vorhanden ist. Auch die Swish-E-Autoren empfehlen Libxml2 zu verwenden, da die internen Parser etwas magerer ausgestattet sind.

Kommentierte Bilder: Jpeg

Auf ähnliche Weise indiziert Swish-E auch Grafikformate: An Jpeg- und Tiff-Bilder lassen sich Metadaten anhängen, im einfachsten Fall schlichte Kommentare. Zum Beispiel fragt Gimp beim Speichern nach einer Textbeschreibung. Wer diese von »Created with the GIMP« in etwas Hilfreiches ändert, findet seine Bilder auch in großen Archiven schnell. Um einen Kommentar aus einer Jpeg-Datei zu extrahieren, stellt die Jpeg Group das Programm »rdjpgcom« [3] zur Verfügung. Es schreibt einfach den Kommentar an Stdout – genau das braucht Swish-E.

Da das Jpeg-Format recht flexibel ist, haben sich die Hersteller von Digitalkameras den so genannten Exif-Header einfallen lassen, der zu einem Foto Informationen wie die Brennweite oder das Kameramodell aufnimmt. Das Programm »jhead« [4] zieht diese Daten aus dem Bild. Es gibt auch den Kommentar zu einem normalen Jpeg-Bild aus, nur beschränkt es sich leider auf eine einzige Zeile. Um beiden Verfahren gerecht zu werden, muss ein kleines Bash-Skript her, siehe Listing 4.

Diese Skript extrahiert zunächst die Informationen per »jhead« und untersucht dann mit Grep, ob darin die Phrase »Camera model :« vorkommt. Nur wenn sie existiert, handelt es sich auch um ein Exif-Foto. Falls nicht, ruft das Skript »rdjpgcom« auf. In der »FileFilter«-Direktive aus Listing 3 ist nun noch der Name »rdjpgcom« durch den Namen dieses Skripts zu ersetzen.

Open-Office-Dokumente durchsuchen

Open Office speichert seine Dateien als Zip-Archive, in denen der eigentliche Inhalt immer in der XML-Datei »content.xml« enthalten ist. Um diese Dokumente zu durchsuchen, ist etwas mehr Aufwand erforderlich. Zunächst soll der Filter für alle Arten von Open-Office-Dateien wirksam sein, also für verschiedene Suffixe gelten.

Die »IndexContents«-Direktive in Listing 5 (Zeile 3) ordnet Texte, Tabellen und Präsentationen dem XML-Format zu. Etwas knifflig fällt die »FileFilterMatch«-Anweisung aus. Sie definiert die Dateitypen über den regulären Ausdruck »/\.(sxw|sxc|sxi)$/i« und ordnet ihnen das Unzip-Programm zu, inklusive der Aufrufparameter »"-p \"%p\" content.xml"«. Damit extrahiert Unzip die Datei »content.xml« und leitet sie an die Standardausgabe weiter.

Eine Besonderheit ist hier die Zeile »StoreDescription«. Eigentlich ist diese Direktive dafür gedacht, kurze Beschreibungstexte in den Index aufzunehmen, die Swish-E bei einer erweiterten Suche anzeigt. Unter anderem ist hier das Tag anzugeben, das die Beschreibung enthält. Optional lässt sich sogar der Umfang der Beschreibung begrenzen. Das hat im Grunde nichts mit der normalen Indizierung eines XML-Dokuments zu tun. Die Praxis zeigt aber, dass Swish-E Open-Office-Dokumente nur dann richtig indiziert, wenn diese Option angegeben ist. Andernfalls bricht der Parser häufig zu früh ab und lässt einen großen Teil des Textes unbeachtet.

Mailbox

Ganz anders funktioniert die Indizierung einer Mailbox. Da Swish-E als Suchresultat immer den Verweis auf eine oder mehrere Dateien liefern, muss jede einzelne Mail als Datei vorliegen. Das ist normalerweise nicht der Fall. Gängige Mailclients – etwa Mozilla Mail – fassen die E-Mail typischerweise in großen Dateien zusammen.

Bei Open-Source-Produkten ist das MBox-Format beliebt, in dem jede Mail mit einer »From«-Zeile beginnt und in ei-

Listing 1: Mini-Konfiguration für HTML-Dateien

```
01 # localweb.conf
02 IndexDir   /srv/www/htdocs
03 IndexOnly  .html
04 IndexFile  ./localweb.index
```

Listing 2: Konfiguration für PDF

```
01 # PDF
02 IndexContents TXT* .pdf
03 FileFilter .pdf "/usr/bin/pdftotext" "'%p' -"
```

Listing 3: Basiskonfiguration für kommentiertes Jpeg

```
01 # JPG
02 IndexContents TXT* .jpg
03 FileFilter rdjpgcom $1
```

Listing 4: Exif- und Jpeg-Kommentare lesen

```
01 #!/bin/bash
02 if jhead $1 | grep "Camera model :" >/dev/null 2>&1; then
03   jhead $1
04 else
05   rdjpgcom $1
06 f
```

Listing 5: Filter für Open Office

```
01 # Open Office
02 FileFilterMatch "/usr/bin/unzip" "-p \"%p\" content.xml"
   /\.(sxw|sxc|sxi)$/i
03 IndexContents XML* .sxw .sxc .sxi
04 StoreDescription XML* <text:p>
```

ner Leerzeile endet. Da auch Mailinglisten-Admins für ihre Archive aus MBox-Dateien viele kleine HTML-Seiten erzeugen wollen, gibt es mehrere geeignete Lösungen. Eine ist Hypermail [5], das ursprünglich ebenfalls vom Swish-Autor Kevin Hughes entwickelt wurde.

```
hypermail -d /home/rolf/mail/Inbox ➥
  -m /srv/www/htdocs/hmail
```

Dieser Aufruf erwartet die E-Mail-Sammlung in der Datei »/home/rolf/mail/Inbox« und schreibt sie ins Verzeichnis »/srv/www/htdocs/hmail«.
Swish-E direkt auf die erzeugten Dateien loszulassen hat einen unerwünschten Nebeneffekt: Indexseiten, die Mails nach Betreff oder Datum sortiert aufführen, erzeugen überflüssige Treffer, da sie die Betreffzeilen sämtlicher Mails enthalten. Wenn der Suchbegriff also in der Betreffzeile auftaucht, muss man sich mindestens durch die vierfache Treffermenge wühlen. Aber das lässt sich mit Swish-E vermeiden: Die Direktive »FileRules« schließt Files von der Indizierung aus. Damit genügen die zwei Zeilen aus Listing 6, um die Hypermail-Version der Mailbox zu indizieren.

Listing 6: Die Indizierung der Mailbox

```
01 IndexDir /srv/www/htdocs/hmail
02 FileRules filename regex
   /(date.html|subject.html|attachment.html|index.html)/
```

Listing 7: Auslesen einer Tabelle

```
01 #!/usr/bin/php -q
02 <?php
03 $dbcon = pg_Connect ("user=ich password=geheim
   dbname=fehler");
04 $qu = pg_exec ($dbcon, "select * from logbuch");
05 for ($row = 0; $row < pg_numRows ($qu); $row++)
06 {
07   $data = pg_fetch_object ($qu, $row);
08   $str = "<html><body>";
09   $str .= $data->eintrag;
10   $str .= "</body></html>\n";
11   $path = "http://localhost/logbuch.php?nummer=$data-
   >nummer";
12   $size = strlen($str);
13   /* Hier beginnt die eigentliche Ausgabe für Swish-E
   */
14   print "    Path-Name: $path\n";
15   print "    Content-Length: $size\n";
16   print "\n";
17   print $str;
18 }
19 pg_freeResult ($qu);
20 pg_close ($dbcon);
21 ?>
```

In der Ferne suchen: Webseiten im Index

Wer Google & Co. Konkurrenz machen möchte, wird nicht alle Daten in lokalen Verzeichnissen finden, sondern muss auf den Transfer per HTTP zurückgreifen. Swish-E beherrscht dies ebenfalls, der Kommandozeilenparameter »-S http« wählt das Hypertext Transfer Protocol aus. Das Paket enthält zusätzlich ein Hilfsprogramm, das sich sinnigerweise »spider.pl« (Spinne) nennt.
Zwar ist die Konfiguration in beiden Fällen recht einfach, aber die Geschwindigkeit lässt sehr zu wünschen übrig. Swish-E holt die einzelnen Seiten sequenziell und legt zwischen den Seiten ein kleines Päuschen ein. Die Einstellung »delay_sec = > 0« streicht zwar diese Pause, die bessere Technik ist aber »keep_alive«-Requests zu aktivieren. So wickelt der Indizierer über eine Verbindung mehrere Requests ab. Falls eine Anfrage hängt, arbeiten die anderen Requests ungestört weiter.
Auch von sich aus optimiert Swish-E den Zugriff: Es lädt jede Seite nur einmal, auch wenn sie durch die Linkstruktur einer Website mehrfach zu erreichen ist. Aber verglichen mit der Indizierung auf der lokalen Festplatte mutet die Webindizierung trotz aller Tricks wie ein Schneckenrennen an.

Datenbanken

Ein ordentlicher Systemadministrator wird alle Änderungen, die er am System vornimmt, in einem Logbuch vermerken. Wenn er sehr viel Zeit hat, wird er dazu vielleicht eine Datenbank einsetzen. Genauer eine Tabelle mit Namen »logbuch«, die aus zwei Feldern besteht: »nummer« für eine eindeutige Nummerierung der Einträge und »eintrag« für die eigentliche Notiz. Um sich das Logbuch anzusehen, benutzt er seinen Webserver und ein PHP-Skript, dem er die Nummer des Eintrags mitgibt. Mit »http://localhost/logbuch.php?nummer = 12« gelangt er an Eintrag 12.
Dieses Beispiel ist zwar an den Haaren herbeigezogen, aber es illustriert etwas Alltägliches: Datenbanken im Internet und Intranet. Swish-E ergänzt die normale Abfrage einer Datenbank ohne Schwierigkeiten um eine Volltextsuche. Natürlich ließe sich diese Aufgabe auch mit dem »spider.pl«-Skript bewerkstelligen, es geht bei gleichem Aufwand aber erheblich schneller.
Die Swish-E-Distribution enthält ein Datenbank-Beispiel, allerdings ist es in Perl geschrieben. Wer lieber PHP nutzt, findet in Listing 7 ein passendes Skript. In der ersten Zeile ist PHP als Kommandoprozessor angegeben, zudem muss das Programm Execute-Rechte besitzen. Der Parameter »-q« sorgt dafür, dass PHP den Output nicht um zusätzliche Header-Direktiven erweitert. Ab Zeile 3 öffnet das Skript eine Verbindung zur PostgreSQL-Datenbank und wählt per SQL-»select«-Kommando den kompletten Inhalt der Tabelle »logbuch«. Die eigentliche Arbeit wird in der Schleife durchgeführt.
Für jeden Datensatz erzeugt das Skript eine HTML-Seite und gibt sie mit »print« an Stdout aus. Damit geht das Programm ähnlich vor wie die Filter der anderen Fremdformate. Neu sind die beiden Schlüsselwörter »Path-Name« und »Content-Length«, die vom eigentlichen Inhalt durch eine Leerzeile getrennt sind. Sie untergliedern den eventuell riesigen Output in kleine Häppchen, die Swish-E unter dem in »Path-Name« angegeben Namen registriert – hier die Webseite, über die der jeweilige Logbucheintrag zu erreichen ist. Das Feld »Content-Length« enthält den Umfang des jeweiligen Häppchens.

Metanamen aus Pfad und HTML-Metatags

In einem gut strukturierten Verzeichnisbaum sind die Pfadangaben eine gute Hilfe, um die Datei-Inhalte zu kategorisieren. Alles unter »~/Bilder/Foto« sollte dann ein Bild sein, konkreter ein Foto. Diese Einteilung ist auch ein guter Ansatzpunkt, um eine Suche nur auf Bilder einzuschränken. In Swish-E sind dafür »MetaNames« zuständig.
Ursprünglich stammen Metanamen aus dem Fundus der HTML-Tags. Swish-E kann auch sie extrahieren und nutzbar

machen. Dazu muss der Indizierer zunächst wissen, welche Metatags er auswerten soll: Die »MetaNames«-Direktive erledigt das. So lassen sich mit dem Eintrag »MetaNames bereich« die eigenen HTML-Dateien klassifizieren, wenn man sie um ein Tag der Form »<meta name="bereich" content="Programmierung">« ergänzt.

Um auch die Pfadangaben zu berücksichtigen, bringt Swish-E ein vordefiniertes Schlüsselwort mit: »swishdocpath«. Mit der Direktive »MetaNames swishdocpath« nimmt es den kompletten Pfad zum Dokument als Schlüsselwort mit in den Index auf. Eine Suche lässt sich dann auf Dateien einschränken, die sich in einem bestimmten Verzeichnis befinden oder bei denen eine bestimmte Zeichenfolge in der Pfadangabe auftaucht. So sucht »swish-e -w Dortmund swishdocpath = (Foto)« alle Dokumente, in denen das Wort Dortmund auftaucht und die in einem Pfad abgelegt sind, der das Wort Foto enthält.

Bequemes Webinterface mitgeliefert

Wer gerne bequemer suchen möchte als mit der Kommandozeile, wird in dem Verzeichnis »example« in der Swish-E-Distribution fündig. Dort wartet ein vollständiges Webinterface auf seine Installation. Es muss im »cgi-bin«-Verzeichnis des Webservers liegen und ist nach ein paar Anpassungen in der Datei »swish.cgi« betriebsbereit. Das Ergebnis einer Suche wird dann wie in Abbildung 1 präsentiert. In der Ergebnisseite hebt das CGI-Skript die Suchbegriffe optisch hervor und zeigt zudem noch einige Metadaten an, etwa die Dateigröße.

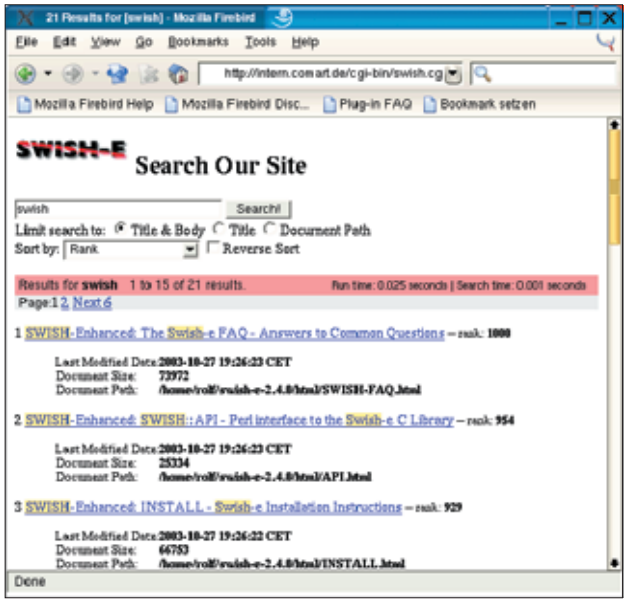

Abbildung 1: Swish-E verbindet Indizierer und Suchmaschine. Hier sucht jemand über das Webinterface das Wort »Swish« innerhalb der Swish-E-Dokumentation.

Beim Suchen mit dem Browser kann ein neues Problem auftreten: Wenn das Indizieren direkt über das Dateisystem läuft, beziehen sich auch die Pfadangaben im Index auf das Wurzelverzeichnis. Dagegen erwartet der Webserver die Angaben relativ zu seinem Dokumenten-Stammverzeichnis. Solche Pfadumformungen erledigt die Direktive »ReplaceRules«. Die Rules suchen nach bekannten Pfadstücken bei den analysierten Dateien und ersetzen sie durch neue. Die Regel »ReplaceRules replace /srv/www/htdocs http://localhost« ersetzt den Original-Pfadnamen »/srv/www/htdocs/index.html« durch die URL »http://localhost/index.html«. Mit dieser sollten weder Server noch Browser Probleme haben.

Gefunden

Mit Swish-E steht ein leistungsfähiges Programmpaket zur Verfügung, das den Inhalt von Dateisystemen oder Websites indiziert. Da sich das Tool auch in eigene Programme einbinden lässt (siehe Kasten „GUI-Suche"), kann jeder Entwickler seiner Software eine ausgereifte Volltextsuche mitgeben. *(fjl)* ■

GUI-Suche

Der Autor dieses Artikels wollte auf seinem Notebook nicht extra einen Webserver anwerfen, nur um ein paar Dateien zu suchen. Die Kommandozeile ist auch nicht jedermanns Sache. Daher hat er ein eigenes GUI-Frontend entwickelt.

Zwar bietet Swish-E sowohl eine C- als auch eine Perl-Schnittstelle [6], die den kompletten Funktionsumfang zur Verfügung stellen. Einfacher war es aber, innerhalb des GUI-Programms Swish-E in einer Shell zu starten und den Output auszuwerten. Das Ergebnis ist ein kleines Programm [7], das die Suchbegriffe abfragt und das Ergebnis in einer Liste anzeigt (Abbildung 2). Ein Klick öffnet jedes Dokument aus der Liste.

In der Konfiguration kann der Benutzer mehrere Suchindizes auswählen und zusätzlich aktivieren. So ist es möglich, die Pfade zu verschiedenen Indizes festzulegen (zum Beispiel Notebook oder Server) und je nach Anwendungsfall einzelne auszublenden. Sehr praktisch, wenn kein Zugriff auf den Server-Index möglich ist, da man gerade unterwegs ist.

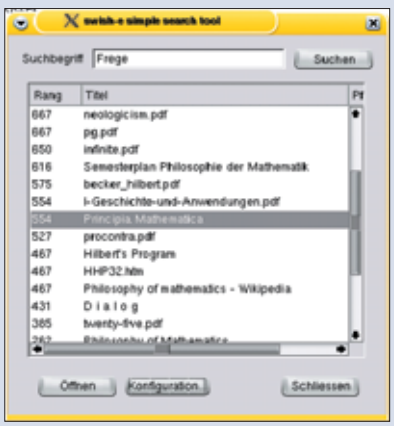

Abbildung 2: Die Oberfläche Ssst (Swish-E Simple Search Tool) sucht nach dem Philosophen Frege in einer Sammlung philosophischer Texte.

Infos

[1] Swish-E: [http://www.swish-e.org]
[2] Pdftotext (Teil der XPdf-Programmsammlung): [http://www.foolabs.com/xpdf/]
[3] Jpeg-Kommentare extrahieren: [ftp://ftp.uu.net/graphics/jpeg/jpegsrc.v6b.tar.gz]
[4] Exif-Header extrahieren: [http://www.sentex.net/~mwandel/jhead/]
[5] Hypermail: [http://www.hypermail.org/]
[6] Michael Schilli, „Perlensuche": Linux-Magazin 10/03, S. 93
[7] Ssst: [http://www.scaldra.net/de/ssst.php]

Der Autor

Rolf Strathewerd ist Gründer und Geschäftsführer der Com Art Software GmbH, schreibt momentan seine BA-Arbeit in Philosophie und ist (natürlich) Linux-Fan.