Finding dead web links with Linkchecker

MISSING LINK

Linkchecker grooms your site to uncover references to missing pages.

BY HAGEN HÖPFNER

If you manage a big Internet site, you are probably familiar with the problem of dead links. An external page at the end of a link can easily go offline, and when a user clicks the link, the browser jumps headlong into a black hole. This problem can be difficult to catch, unless you plan to spend all your spare time testing links at your own website. If you're looking for a faster way to hunt for holes, try Linkchecker.

THE AUTHOR

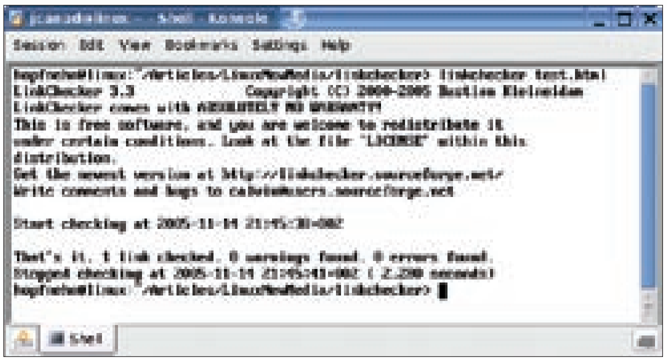
Hagen Höpfner has now completed his doctorate in computer science, and is now a lecturer for database and information systems at the International University in Germany (*http://www.i-u.de*), Bruchsal, Germany. Besides being a loving father, Hagen is the co-author of a reference manual on mobile databases and information systems. In his leisure time, he plays guitar with a rock band called "Gute Frage" (*http://www.gutefrage.info*).

The easiest way to install Linkchecker [1] is to use a prebuilt version. For RPM-based systems such as Suse or Fedora, an RPM package with the current version 3.3 and a source code archive is available from [2]. Users with Debian derivatives should check out the site at [3] or install Linkchecker by running *apt-get install linkchecker*.

Gap Searching

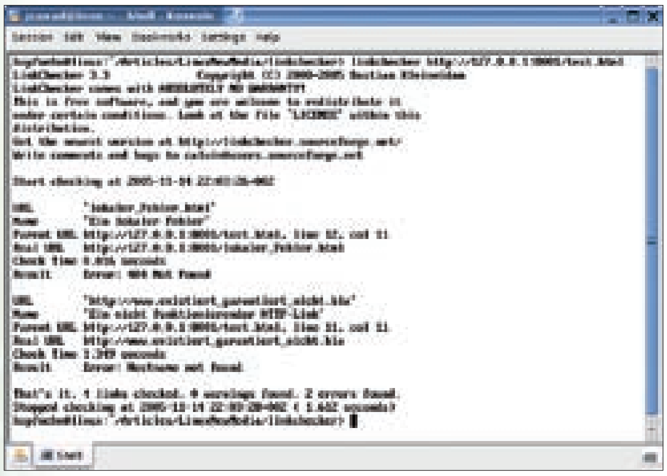
Linkchecker runs in the command line based on the following pattern *linkchecker options test_object*, where *test_object* is a local or a remote file accessible via HTTP or FTP. An entry such as *linkchecker www.linux-magazine.com* would work, however, as Linkchecker automatically adds the URL prefixes *http://* or *ftp://*. If you use a proxy to surf the Internet, you will need to tell Linkchecker by setting the environmental variables *http_proxy*, *https_proxy*, *ftp_proxy*, and *gopher_proxy*. The exact syntax depends on your shell. If you have Bash, you can *export http_proxy = "http://localhost:8080"*; those of you with the Tc shell can enter *setenv http_proxy "http://localhost:8080"*. A proxy that listens to port *8080* on *localhost* passes the link check on to the Internet.

A simple, but not quite standards-compliant HTML file titled *test.html* reveals how Linkchecker works: entering *linkchecker test.html* does not seem to work at first. Linkchecker simply reports that it has checked a link and not found any errors (Figure 1). However, the test



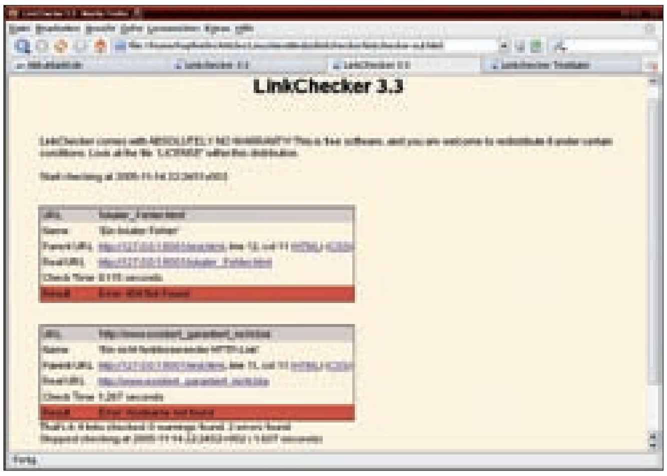
```
hoepfner@linux:~/Articles/LinuxNewMedia/linkchecker> linkchecker test.html
LinkChecker 3.3          Copyright (C) 2000-2005 Bastian Kleineidam
LinkChecker comes with ABSOLUTELY NO WARRANTY!
This is free software, and you are welcome to redistribute it
under certain conditions. Look at the file `LICENSE' within this
distribution.
Get the newest version at http://linkchecker.sourceforge.net/
Write comments and bugs to calvin@users.sourceforge.net

Start checking at 2005-11-14 21:45:38+002

That's it. 1 link checked. 0 warnings found. 0 errors found.
Stopped checking at 2005-11-14 21:45:41+002 (2.290 seconds)
hoepfner@linux:~/Articles/LinuxNewMedia/linkchecker> █
```

Figure 1: Linkchecker has problems with the filesystem. It ignores two errors and just tests one link.



```
[illegible] linkchecker http://127.0.0.1:8000/test.html
LinkChecker 3.3            Copyright (C) 2000-2005 Bastian Kleineidam
LinkChecker comes with ABSOLUTELY NO WARRANTY!
This is free software, and you are welcome to redistribute it
under certain conditions. Look at the file `LICENSE' within this
distribution.
Get the newest version at http://linkchecker.sourceforge.net/
Write comments and bugs to calvin@users.sourceforge.net

Start checking at 2005-11-14 22:03:26+002

URL        'lokaler_Fehler.html'
Name       'Ein lokaler Fehler'
Parent URL http://127.0.0.1:8000/test.html, line 12, col 11
Real URL   http://127.0.0.1:8000/lokaler_Fehler.html
Check time 0.036 seconds
Result     Error: 404 Not Found

URL        'http://www.existiert_garantiert_nicht.biz'
Name       'Ein nicht funktionierender HTTP-Link'
Parent URL http://127.0.0.1:8000/test.html, line 21, col 11
Real URL   http://www.existiert_garantiert_nicht.biz
Check time 1.289 seconds
Result     Error: Hostname not found

That's it. 4 links checked. 0 warnings found. 2 errors found.
Stopped checking at 2005-11-14 22:03:28+002 (1.662 seconds)
[illegible]
```

Figure 2: Finding faults: Linkchecker finds the errors in "test.html" with some help from the web server. The tool checks four links and reports that two are bad.



LinkChecker 3.3

[illegible]

URL	[illegible]
Name	[illegible]
Parent URL	[illegible]
Real URL	[illegible]
Check Time	[illegible]
Result	Error: 404 Not Found

URL	[illegible]
Name	[illegible]
Parent URL	[illegible]
Real URL	[illegible]
Check Time	1.287 seconds
Result	Error: Hostname not found

Figure 3: Outputting the results to an HTML document gives you links in the HTML output, which you can follow to double check the results.

file actually includes three links, two of which point to a black hole.

Interestingly, if the *test.html* file is served up by a web server, Linkchecker finds the faulty links (Figure 2). The documentation also states that the current version does not support JavaScript.

The Right Pattern

In addition to text output, you can select one of the formats listed in Table 1. Depending on whether you need screen output or a special output file, you can stipulate *--output = html* or *--file-output = html/filename* for HTML format. If you don't specify an output file name, Linkchecker calls the file *linkchecker-out* and adds the file type, for example *linkchecker-out.html*. Figure 3 shows the browser view of an output file: the links, which allow you to recheck the results of the search, are a useful aid.

The output is sent to the terminal at the same time as to the file. If you want Linkchecker to run in the background, set the *-q* option, which suppresses the command line output. Some output formats, such as *gml*, additionally require the *--verbose* option. The option tells Linkchecker to add information about working links to the graph.

Fine Tuning

Large Internet sites typically comprise a number of interrelated files. Linkchecker begins its search at the start address and recursively searches the whole domain. Typing *linkchecker www.linux-magazine.com* tells the program to check all pages that start with *http://www.linux-magazine.com*.

The *-r* option sets the recursion depth. *linkchecker -r1 www.linux-magazine.com* checks the welcome page and all pages that link from this page. If you specify *-r2* instead, the program delves one level deeper. The *--no-follow-url = regex* and *--ignore-url = regex* options also let you influence automatic link checking. The former case tells Linkchecker to analyze URLs that match the regular expression, *regex*, but not to search the subpages. You can also specify *--ignore-url URLs* to define web pages that Linkchecker should ignore.

Table 1: Output Foramts

Keyword	Meaning
text	Standard text output to a flat text file
html	Output as HTML code containing links to the referenced pages
csv	Output as comma-separated text with one URL per line.
gml	Output as GML-formatted graph [6]
dot	Output as .dot-formatted graph [7]
gxml	Output in GraphXML format [4]. Viewers are available here [5]
xml	Machine readable: output in XML format
sql	Output as INSERT instruction for SQL
blacklist	This option tells Linkchecker to log error information only, and write this information to *~/.linkchecker/blacklist*. This is useful for automating link checks with cron.
none	No output, useful for scripting

For reasons of efficiency, Linkchecker investigates multiple files in parallel, typically up to ten files at the same time. You can change the maximum number of links to check by setting the *-tX* option, where *X* represents the number. The *man linkchecker* command gives you a comprehensive list of options.

Conclusions

Linkchecker is a useful tool for discovering dead links. The variety of output formats allows for structured processing of any link errors the process discovers. In our lab, Linkchecker had trouble with files on its own filesystem, but nowhere else. ■

INFO

[1] Linkchecker homepage: *http://linkchecker.sourceforge.net/*

[2] Linkchecker download: *http://sourceforge.net/project/showfiles.php?group_id=1913*

[3] Linkchecker for Debian: *http://packages.debian.org/unstable/web/linkchecker*

[4] Ivan Herman, M. Scott Marshall: GraphXML – An XML-Based Graph Interchange Format: *http://db.cwi.nl/rapporten/abstract.php?abstractnr=613*

[5] GVF – The Graph Visualization Framework: *http://gvf.sourceforge.net/*

[6] The GML file format: *http://infosun.fmi.uni-passau.de/Graphlet/GML/*

[7] Graphviz – Graph Visualization Software for displaying DOT files: *http://www.research.att.com/sw/tools/graphviz/*