



by Heinz-Josef Claes
<hjclaes/at/web.de>

storeBackup, das unkonventionelle Backup Tool



Abstract:

StoreBackup bietet sich für Privatpersonen an, die zwar nicht über ein Bandgerät, aber eine zweite Festplatte oder über einen weiteren Rechner verfügen. Es bietet sich im professionellen Umfeld an, um den Anwendern einen extrem schnellen und bequemen Zugriff auf ihre Backups zu ermöglichen, um Kosten für Bänder und um Administrationskosten einzusparen.

Als Alternative oder als Ergänzung zu einer Datensicherung auf Band bietet sich die Speicherung auf Festplatten oder vergleichbare Medien an. Das hier vorgestellte Programm ermöglicht dieses auf sehr performante und Speicherplatz sparende Art und Weise:

- Verzeichnisse werden rekursiv an eine andere Stelle kopiert (z.B. /home => /var/bkup/2003.12.13_02.04.26). Hierbei bleiben die Rechte der Dateien erhalten, so dass Anwender direkt auf das Backup zugreifen können.
- Die Dateien werden auf ihren Inhalt untersucht. Es wird sichergestellt, dass der Inhalt jeder Datei nur einmal im Backup gespeichert wird, d.h. Dateien gleichen Inhalts existieren physisch nur einmal im Backup.
- Gleiche Dateien werden über hard links miteinander verbunden und erscheinen so im Backup an denselben Stellen wie im Original
- Dateien im Backup werden komprimiert, sofern dieses nicht über exclude-Pattern ausgeschlossen wird. Die Kompression kann auch

vollständig unterbunden werden.

- Voneinander unabhängig erstellte Backupreihen (z.B. von unterschiedlichen Rechnern) können über hard links auf gemeinsame Dateien verweisen. Auch können mit dieser Methode konfigurierbar vollständige und teilweise Backups erstellt werden – immer unter der Prämisse, dass Dateien gleichen Inhalts nur einmal physisch im Backup vorkommen.
-

Warum ein neues Backup Tool?

Es gibt wahrscheinlich tausende von Backup Programmen. Warum also noch ein weiteres? Der Anlass ergab sich für mich aus meiner Tätigkeit als Berater. Ich war die ganze Woche unterwegs und hatte keine Möglichkeit, die Daten meines Laptops während der Woche zu Hause zu sichern. Alles, was ich zur Verfügung hatte, war ein 250 MByte ZIP–Laufwerk an der parallelen Schnittstelle. Um ein Backup auf dem ZIP–Laufwerk zu erstellen, hatte ich daher relativ wenig Platz zur Verfügung und musste mit einer schlechten Performance (ca. 200 KByte/s) leben. Außerdem wollte ich einen schnellen und einfachen Zugriff auf meine Dateien – die üblichen Varianten mit full, differential und incremental Backups (z.B. mit tar oder dump) gefiel mir nicht: Zum einen ist es meist mühsam, eine Datei in der gewünschten Version wiederzufinden und zum anderen kann man nicht beliebig alte Sicherungen löschen, sondern muss dieses beim Erstellen des Backups sorgfältig planen.

Mein Ziel war es, während der Arbeit 'mal schnell ein Backup machen zu können und die gesuchten Dateien schnell und unproblematisch wiederfinden zu können.

So entstand Ende 1999 die erste Version von storeBackup, die jedoch bei Einsatz in größeren Umgebungen nicht geeignet war. Sie war nicht performant genug, skalierte nicht hinreichend und war nicht in der Lage, mit unangenehmen Dateinamen (z.B. '\n' im Namen) umzugehen.

Mit den Erfahrungen dieser ersten Version habe ich dann eine neue geschrieben, die ich dann ein knappes Jahr später unter der GPL veröffentlicht habe. Inzwischen gibt es einen nicht mehr so kleinen Anwenderkreis, der storeBackup von der privaten Datensicherung, der Sicherung von (Mail–) Verzeichnissen bei ISPs oder Kliniken und Universitäten bis hin zur Archivierung nutzt.

Was wäre ein ideales Backup Tool?

Das ideale Backup Tool mit minimalem Aufwand für die Administration und maximalen Komfort für den Anwender würde von jedem Tag eine komplette Kopie des gesamten Datenbestandes (mit den jeweiligen Zugriffsrechten) auf ein anderes Dateisystem machen. Die hierzu benötigten Rechner und Festplattensysteme sollten selbstverständlich in einem entfernten, entsprechend gesicherten Gebäude untergebracht sein. Der Anwender könnte dann z.B. über einen Dateisystem Browser auf diese gesicherten Daten zugreifen, um in ihnen zu suchen oder die benötigten Dateien direkt zurückzukopieren. Das Backup würde so zu etwas direkt und problemlos verwendbarem. Hierdurch würde der Umgang mit Backups – da der Umweg über die Administration in der Regel nicht mehr notwendig ist – zu etwas "normalem".

Das hier beschriebene Vorgehen hat leider einen kleinen Nachteil: Es benötigt extrem viel Plattenplatz und ist – da jeweils die gesamte Datenmenge kopiert werden muss – auch ziemlich langsam.

Wie funktioniert storeBackup?

StoreBackup versucht, das oben beschriebene "ideale Backup" zu realisieren und die beiden Problem – Plattenplatz und Performance – zu lösen.

Features

Die erste Massnahme zur Senkung des verbrauchten Plattenplatzes ist, die Dateien – wenn sinnvoll – zu komprimieren. StoreBackup ermöglicht hierbei die Verwendung eines beliebigen Kompressionsalgorithmus, der als externes Programm vorliegen muss. Defaultmässig wird bzip2 verwendet.

Bei näherer Betrachtung der gespeicherten Dateien fällt auf, dass sich von Backup zu Backup nur relativ wenige Dateien ändern – aus diesem Grund werden schließlich auch inkrementelle Backups durchgeführt. Es fällt aber auch auf, dass viele Dateien gleichen Inhalts innerhalb eines Backups vorkommen, weil Anwender Dateien kopieren oder z.B. mit einer Versionsverwaltung wie cvs gearbeitet wird. Auch werden Dateien oder ganze Dateibäume vom Anwender umbenannt, die bei einem üblichen inkrementellen Backup dann (unnötigerweise) wiederum gesichert werden. Die Lösung liegt darin, zu untersuchen, ob eine Datei gleichen Inhalts bereits (eventuell komprimiert) im Backup vorliegt und dann auf diese zu verweisen. Dieser Verweis ist über einen Hard Link realisiert. (Erläuterung: Die Blöcke einer Datei werden unter Unix über Inodes verwaltet. Auf einen Inode können viele verschiedene Dateinamen in unterschiedlichen Verzeichnissen verweisen. Die eigentliche Datei wird erst mit dem letzten Hard Link (= Directory Namen) gelöscht. (Hard Links können nur innerhalb eines Filesystems auf dieselbe Datei existieren.))

Über den Trick mit den Hard Links, die bei bereits im Backup vorhandenen Dateien auf diese gesetzt werden, liegen dann in jedem Backup sämtliche Dateien individuell vor, obwohl gleichartige Dateien nur einmal physisch auf der Platte gespeichert werden. Das Kopieren und Umbenennen von Dateien oder ganzen Verzeichnisbäumen kostet so im Backup nur den Platz für die Hard Links – also fast nichts.

Es ist aber üblicherweise so, dass nicht nur ein Rechner gesichert werden soll, sondern mehrere. Diese haben, insbesondere wenn Verzeichnisse wie /etc oder /usr gesichert werden, einen sehr hohen Anteil an identischen Dateien. Es liegt nahe, dass auch hier identische Dateien nur einmal auf der Sicherungsplatte stehen sollten. Die einfachste Lösung, dieses zu erreichen, ist, alle Verzeichnisse von dem Backup Server zu mounten und alle Rechner gemeinsam zu sichern. Doppelte Dateien werden so erkannt und über Hard Links verbunden. Dieses Verfahren hat allerdings den Nachteil, dass alle zu sichernden Rechner gleichzeitig für die gesamte Zeit des Backups zur Verfügung stehen müssen. In vielen Fällen, z.B. wenn Notebooks mit storeBackup gesichert werden sollen, ist ein solches Vorgehen nicht realisierbar.

Jedoch gerade bei Notebooks ist, da üblicherweise Dateien von den Anwendern lokal kopiert werden, ein sehr hoher Überdeckungsgrad der Dateien der Anwender gegeben. Für solche Fälle, oder wenn Server unabhängig voneinander gesichert werden, aber trotzdem der verfügbare Plattenplatz über Hard Links optimal genutzt werden soll, kann storeBackup auch in unabhängige Backupreihen (d.h. unabhängig voneinander (von eventuell unterschiedlichen Rechnern) erstellte Backups) die Dateien über Hard Links miteinander verbinden.

Zum Löschen der Dateien bietet storeBackup einen Satz von unterschiedlichen Regeln an. Der große Vorteil beim Löschen ist, dass jedes Backup ein Full Backup ist – es können also beliebige Backups gelöscht werden. Es gibt keine Notwendigkeit – wie bei traditionellen Backups – zu berücksichtigen, dass ein inkrementelles Backup nur dann verwendbar ist, wenn die Vorgänger noch verfügbar sind.

Die Regeln erlauben das Löschen bzw. Erhalten von Sicherungen nach dem Wochentag, ersten oder letzten Tagen in der Woche / im Monat oder im Jahr. Auch kann sichergestellt werden, dass eine minimale Anzahl von Sicherungen behalten wird. Dieses ist insbesondere dann sinnvoll, wenn Backups nicht regelmäßig durchgeführt werden. So können z.B. nach einem 4-wöchigem Urlaub die letzten n Backups des Laptops

erhalten werden, obwohl das Erhalten von Backups nur auf drei Wochen eingestellt war. Es ist weiterhin möglich, eine maximale Anzahl von Backups anzugeben. Hier existieren weitere Möglichkeiten um Konflikte bei divergierenden Regeln (sinnvoll) zu lösen.

Performance

Das oben beschriebene Verfahren geht davon aus, dass überprüft wird, ob zu einer zu sichernden Datei bereits eine gleichen Inhalts im Backup vorliegt. Dieses betrifft sowohl Dateien in der alten Sicherung, als auch im gerade angelegten neuen Backup. Es macht natürlich keinen Sinn, jede zu sichernde Datei mit jeder bereits gesicherten zu vergleichen. Daher werden von den gesicherten Dateien die md5 Summen vorgehalten, mit denen die md5 Summe der zu sichernden Datei unter Verwendung einer Hash-Tabelle verglichen wird. Das Programm verwendet hierzu dbm Files.

Das Berechnen der md5 Summe geht zwar schnell, dauert aber bei größeren Datenmengen auch lange. Aus diesem Grund überprüft storeBackup zuerst, ob die Datei seit der letzten Sicherung unverändert ist (Pfad + Dateiname, ctime, mtime und size gleich). Wenn ja, wird einfach die md5 Summe der letzten Sicherung übernommen und der Hard Link gesetzt. Wenn nicht, wird die md5 Summe berechnet und anschließend überprüft, ob es bereits eine Datei mit derselben md5 Summe existiert. (Beim Abgleich mit mehreren Backup-Serien ist das Verfahren etwas erweitert, aber ähnlich performant.) Durch diese Vorgehensweise müssen für ein Backup in der Regel nur sehr wenige md5 Summen berechnet werden.

Mein Server (200 MHz, IDE) schafft etwas 20–35 files/sec, meine Desktop Maschine (800 MHz, IDE) etwa 150–200 files/sec. Mit einem schnellen Rechner und schnellen Platten (2.4 GHz, 1.4 TB Software RAID 5) habe ich bereits bis zu 800 files/sec gemessen. Diese Werte gelten, wenn auf die lokale Platte geschrieben wird. Beim Schreiben über NFS sind die Werte niedriger. Ausschlaggebend ist insbesondere die Geschwindigkeit der Festplatten. (Alle Tests wurden mit Linux durchgeführt.)

Realisierung

Die storeBackup Tools sind bisher auf den Betriebssystemen Linux, FreeBSD, Solaris und AIX getestet worden. Sie sollten aber auf allen Unix-Plattformen laufen. Als Programmiersprache wird perl verwendet.

Installation

Die Installation ist einfach. StoreBackup kann von www.sourceforge.net/projects/storebackup als storeBackup-version.tar.bz2 heruntergeladen werden. Anschließend kann es an einer gewünschten Stelle ausgepackt werden:

```
tar jxf storeBackup-version.tar.bz2
```

Hierbei wird das Verzeichnis storeBackup erzeugt. In ihm sind die Dokumentation und im Unterverzeichnis bin die ausführbaren Programme. Sie können mit vollständigem Pfad aufgerufen werden. Alternativ kann auch die \$PATH Environmentvariable gesetzt werden. Auf Betriebssystemen, die das Programm md5sum nicht mitliefern (z.B. FreeBSD) muss dieses noch kompiliert werden. Eine Anleitung hierzu steht in der mitgelieferten README Datei.

Bedienung

Hier sollen nicht alle Möglichkeiten im Detail beschrieben werden; sie sind im Softwarepaket beschrieben.

Die einfachste Art, eine Sicherung zu erzeugen ist:

```
storeBackup.pl -s sourceDir -t targetDir
```

Hierbei müssen sourceDir und targetDir existieren. StoreBackup wird die Dateien von sourceDir nach targetDir/datum_uhrzeit kopieren, dabei die Dateien, nicht auf .gz, .bz2, .png etc. enden, mit bzip2 komprimieren und doppelt vorkommende Dateien verlinken.

Das Sicherungsprogramm storeBackup.pl verfügt in der aktuellen Version (1.14.1) über 45 Parameter, die zu beschreiben hier den Rahmen sprengen würde. Sie sind mittels

```
storeBackup.pl -h
```

abrufbar. In den Dateien README und EXAMPLES finden sich umfangreiche Erläuterungen über die unterschiedlichen Anwendungsmöglichkeiten. Hier soll nur darauf hingewiesen werden, dass alternativ zu den Parametern in der Kommandozeile, die schnell unübersichtlich werden können, auch eine Konfigurationsdatei verwendet werden kann. Diese kann mit

```
storeBackup.pl --generate --file ConfigFile
```

oder kürzer mit

```
storeBackup.pl -g -f ConfigFile
```

generiert werden. Wenn die Konfiguration beendet ist, kann dieses mittels

```
storeBackup.pl -f ConfigFile --print
```

gelesen, auf Syntax überprüft und bereits teilweise ausgewertet werden. Anschließend kann storeBackup mittels

```
storeBackup.pl -f ConfigFile
```

gestartet werden. Die vollständige Beschreibung zu allen Optionen von storeBackup befindet sich in den Dateien README und EXAMPLES, die im tar File enthalten sind.

Um festzustellen, wo welche Version einer gesicherten Datei im Backup existiert, kann storeBackupVersion verwendet werden:

```
storeBackupVersion.pl -f Filename
```

Filename ist dabei der Dateiname der interessierenden Datei, und zwar so, wie er im Backup steht, d.h. eventuell mit Kompressionsendung. Am einfachsten geht man in das Backup-Verzeichnis an die betreffende Stelle und ruft das Kommando auf. Mit der Option "-h" wird eine Erläuterung über alle 11 Parameter ausgegeben.

Das Zurückspielen von einzelnen Dateien wird in der Regel mittels cp, ftp, File Browser oder ähnliche Mechanismen erfolgen. Wenn jedoch Teilbäume oder ganze Backups zurückgesichert werden sollen, ist es sinnvoll, das zugehörige Tool storeBackupRecover.pl zu verwenden. Es extrahiert die gewünschten Dateien oder Verzeichnisse aus dem Backup. Hierbei wird der Originalzustand wiederhergestellt, d.h. User, Gruppe und die Rechte werden zurückgesetzt. Die Dateien werden ebenfalls entkomprimiert, sofern sie es im Originalzustand waren. Auch werden hard links, die im Original bestanden, wiederhergestellt. Weitere Optionen von storeBackupRecover erlauben statistische Ausgaben, die Beeinflussung von Performanceparametern, das Überschreibverhalten und anderes. Die insgesamt 10 Parameter können mit Hilfe

der Option '-h' ausgegeben werden.

Mittels storeBackupDel.pl können Backups unabhängig vom Programm storeBackup.pl gelöscht werden. Dieses kann nützlich sein, wenn über NFS gesichert wird. Das Löschen großer Verzeichnisbäume über NFS ist wesentlich langsamer als lokal. In einem solchen Fall kann storeBackup ohne Löschfunktion über NFS aufgerufen werden, was auch die Backupdauer kalkulierbarer macht. Das Löschen von alten Backups auf dem Server kann dann mittels storeBackupDel – welches über dieselben Möglichkeiten zum Löschen wie storeBackup verfügt – vom eigentlichen Backup entkoppelt erfolgen.

Bereits erstellte Backups sind in Verzeichnissen organisiert. Diese können mit storeBackuppls.pl (verständlicher als mittels 'ls') angezeigt werden. Entweder als einfache Liste

```
hjc@schlappix:~/backup ) storeBackuppls.pl /media/zip/stbu/
 1 Fri May 23 2003 2003.05.23_12.37.53 -156
 2 Fri Jun 06 2003 2003.06.06_14.31.47 -142
 3 Fri Jun 13 2003 2003.06.13_14.17.18 -135
 4 Fri Jun 20 2003 2003.06.20_14.02.35 -128
 5 Fri Jun 27 2003 2003.06.27_14.23.55 -121
 6 Mon Jun 30 2003 2003.06.30_17.34.37 -118
 7 Fri Jul 04 2003 2003.07.04_13.10.06 -114
 8 Fri Jul 11 2003 2003.07.11_13.13.14 -107
 9 Fri Jul 18 2003 2003.07.18_14.03.49 -100
10 Fri Jul 25 2003 2003.07.25_14.19.19 -93
11 Thu Jul 31 2003 2003.07.31_17.07.55 -87
12 Fri Aug 01 2003 2003.08.01_12.16.58 -86
13 Fri Aug 15 2003 2003.08.15_15.10.19 -72
14 Sat Aug 23 2003 2003.08.23_06.25.35 -64
15 Wed Aug 27 2003 2003.08.27_18.21.09 -60
16 Thu Aug 28 2003 2003.08.28_14.16.39 -59
17 Fri Aug 29 2003 2003.08.29_14.35.10 -58
18 Mon Sep 01 2003 2003.09.01_17.19.56 -55
19 Tue Sep 02 2003 2003.09.02_18.18.46 -54
20 Wed Sep 03 2003 2003.09.03_16.22.41 -53
21 Thu Sep 04 2003 2003.09.04_16.59.19 -52
22 Fri Sep 05 2003 2003.09.05_14.35.20 -51
23 Mon Sep 08 2003 2003.09.08_20.08.52 -48
24 Tue Sep 09 2003 2003.09.09_18.45.48 -47
25 Wed Sep 10 2003 2003.09.10_18.30.48 -46
26 Thu Sep 11 2003 2003.09.11_17.26.46 -45
27 Fri Sep 12 2003 2003.09.12_15.23.03 -44
28 Mon Sep 15 2003 2003.09.15_18.05.19 -41
29 Tue Sep 16 2003 2003.09.16_18.04.16 -40
30 Wed Sep 17 2003 2003.09.17_19.03.02 -39
31 Thu Sep 18 2003 2003.09.18_18.21.09 -38
32 Fri Sep 19 2003 2003.09.19_14.48.05 -37 not finished
33 Mon Sep 22 2003 2003.09.22_18.58.55 -34
34 Tue Sep 23 2003 2003.09.23_18.48.40 -33
35 Wed Sep 24 2003 2003.09.24_19.32.24 -32
36 Thu Sep 25 2003 2003.09.25_18.05.38 -31
37 Fri Sep 26 2003 2003.09.26_14.59.59 -30
38 Mon Sep 29 2003 2003.09.29_18.42.59 -27
39 Tue Sep 30 2003 2003.09.30_18.02.03 -26
40 Wed Oct 01 2003 2003.10.01_17.09.43 -25
41 Thu Oct 02 2003 2003.10.02_15.26.33 -24
42 Mon Oct 06 2003 2003.10.06_20.08.45 -20
43 Tue Oct 07 2003 2003.10.07_19.46.54 -19
44 Wed Oct 08 2003 2003.10.08_16.03.23 -18
45 Thu Oct 09 2003 2003.10.09_16.58.28 -17
46 Fri Oct 10 2003 2003.10.10_14.21.06 -16
47 Mon Oct 13 2003 2003.10.13_18.58.24 -13
48 Tue Oct 14 2003 2003.10.14_16.02.44 -12
49 Wed Oct 15 2003 2003.10.15_19.04.12 -11
```

50	Thu	Oct	16	2003	2003.10.16_15.47.51	-10
51	Mon	Oct	20	2003	2003.10.20_09.34.52	-6
52	Mon	Oct	20	2003	2003.10.20_12.16.40	-6
53	Tue	Oct	21	2003	2003.10.21_09.43.40	-5
54	Tue	Oct	21	2003	2003.10.21_11.22.36	-5
55	Tue	Oct	21	2003	2003.10.21_16.01.15	-5
56	Tue	Oct	21	2003	2003.10.21_18.08.07	-5
57	Wed	Oct	22	2003	2003.10.22_10.02.51	-4
58	Wed	Oct	22	2003	2003.10.22_16.09.42	-4
59	Wed	Oct	22	2003	2003.10.22_18.03.05	-4
60	Thu	Oct	23	2003	2003.10.23_08.18.15	-3
61	Thu	Oct	23	2003	2003.10.23_14.16.24	-3
62	Thu	Oct	23	2003	2003.10.23_17.00.36	-3
63	Fri	Oct	24	2003	2003.10.24_13.29.30	-2
64	Sun	Oct	26	2003	2003.10.26_09.08.55	0

(Das 'not finished' zeigt hier an, dass das entsprechende Backup abgebrochen wurde.)
oder mit den Informationen zum in der Konfigurationsdatei angegebenen Löschverhalten:

```
hjc@schlappix:~/backup ) storeBackups.pl -f stbu.conf /media/zip/stbu/
analyse of old Backups in </media/zip/stbu/>:
Fri 2003.05.23_12.37.53 (156): keepLastOfMonth(400d)
Fri 2003.06.06_14.31.47 (142): keepLastOfWeek(150d)
Fri 2003.06.13_14.17.18 (135): keepLastOfWeek(150d)
Fri 2003.06.20_14.02.35 (128): keepLastOfWeek(150d)
Fri 2003.06.27_14.23.55 (121): keepLastOfWeek(150d)
Mon 2003.06.30_17.34.37 (118): keepLastOfMonth(400d)
Fri 2003.07.04_13.10.06 (114): keepLastOfWeek(150d), keepMinNumber50
Fri 2003.07.11_13.13.14 (107): keepLastOfWeek(150d), keepMinNumber49
Fri 2003.07.18_14.03.49 (100): keepLastOfWeek(150d), keepMinNumber48
Fri 2003.07.25_14.19.19 (93): keepLastOfWeek(150d), keepMinNumber47
Thu 2003.07.31_17.07.55 (87): keepLastOfMonth(400d), keepMinNumber46
Fri 2003.08.01_12.16.58 (86): keepLastOfWeek(150d), keepMinNumber45
Fri 2003.08.15_15.10.19 (72): keepLastOfWeek(150d), keepMinNumber44
Sat 2003.08.23_06.25.35 (64): keepLastOfWeek(150d), keepMinNumber43
Wed 2003.08.27_18.21.09 (60): keepMinNumber42, keepWeekDays(60d)
Thu 2003.08.28_14.16.39 (59): keepMinNumber41, keepWeekDays(60d)
Fri 2003.08.29_14.35.10 (58): keepLastOfMonth(400d), keepLastOfWeek(150d),
keepMinNumber40, keepWeekDays(60d)
Mon 2003.09.01_17.19.56 (55): keepMinNumber39, keepWeekDays(60d)
Tue 2003.09.02_18.18.46 (54): keepMinNumber38, keepWeekDays(60d)
Wed 2003.09.03_16.22.41 (53): keepMinNumber37, keepWeekDays(60d)
Thu 2003.09.04_16.59.19 (52): keepMinNumber36, keepWeekDays(60d)
Fri 2003.09.05_14.35.20 (51): keepLastOfWeek(150d), keepMinNumber35, keepWeekDays(60d)
Mon 2003.09.08_20.08.52 (48): keepMinNumber34, keepWeekDays(60d)
Tue 2003.09.09_18.45.48 (47): keepMinNumber33, keepWeekDays(60d)
Wed 2003.09.10_18.30.48 (46): keepMinNumber32, keepWeekDays(60d)
Thu 2003.09.11_17.26.46 (45): keepMinNumber31, keepWeekDays(60d)
Fri 2003.09.12_15.23.03 (44): keepLastOfWeek(150d), keepMinNumber30, keepWeekDays(60d)
Mon 2003.09.15_18.05.19 (41): keepMinNumber29, keepWeekDays(60d)
Tue 2003.09.16_18.04.16 (40): keepMinNumber28, keepWeekDays(60d)
Wed 2003.09.17_19.03.02 (39): keepMinNumber27, keepWeekDays(60d)
Thu 2003.09.18_18.21.09 (38): keepMinNumber26, keepWeekDays(60d)
Fri 2003.09.19_14.48.05 (37): keepLastOfWeek(150d), keepMinNumber25, keepWeekDays(60d)
Mon 2003.09.22_18.58.55 (34): keepMinNumber24, keepWeekDays(60d)
Tue 2003.09.23_18.48.40 (33): keepMinNumber23, keepWeekDays(60d)
Wed 2003.09.24_19.32.24 (32): keepMinNumber22, keepWeekDays(60d)
Thu 2003.09.25_18.05.38 (31): keepMinNumber21, keepWeekDays(60d)
Fri 2003.09.26_14.59.59 (30): keepLastOfWeek(150d), keepMinNumber20, keepWeekDays(60d)
Mon 2003.09.29_18.42.59 (27): keepMinNumber19, keepWeekDays(60d)
Tue 2003.09.30_18.02.03 (26): keepLastOfMonth(400d), keepMinNumber18, keepWeekDays(60d)
Wed 2003.10.01_17.09.43 (25): keepMinNumber17, keepWeekDays(60d)
Thu 2003.10.02_15.26.33 (24): keepLastOfWeek(150d), keepMinNumber16, keepWeekDays(60d)
```

```

Mon 2003.10.06_20.08.45 (20): keepMinNumber15, keepWeekDays (60d)
Tue 2003.10.07_19.46.54 (19): keepMinNumber14, keepWeekDays (60d)
Wed 2003.10.08_16.03.23 (18): keepMinNumber13, keepWeekDays (60d)
Thu 2003.10.09_16.58.28 (17): keepMinNumber12, keepWeekDays (60d)
Fri 2003.10.10_14.21.06 (16): keepLastOfWeek (150d), keepMinNumber11, keepWeekDays (60d)
Mon 2003.10.13_18.58.24 (13): keepMinNumber10, keepWeekDays (60d)
Tue 2003.10.14_16.02.44 (12): keepMinNumber9, keepWeekDays (60d)
Wed 2003.10.15_19.04.12 (11): keepMinNumber8, keepWeekDays (60d)
Thu 2003.10.16_15.47.51 (10): keepLastOfWeek (150d), keepMinNumber7, keepWeekDays (60d)
Mon 2003.10.20_09.34.52 (6): keepDuplicate (7d)
Mon 2003.10.20_12.16.40 (6): keepMinNumber6, keepWeekDays (60d)
Tue 2003.10.21_09.43.40 (5): keepDuplicate (7d)
Tue 2003.10.21_11.22.36 (5): keepDuplicate (7d)
Tue 2003.10.21_16.01.15 (5): keepDuplicate (7d)
Tue 2003.10.21_18.08.07 (5): keepMinNumber5, keepWeekDays (60d)
Wed 2003.10.22_10.02.51 (4): keepDuplicate (7d)
Wed 2003.10.22_16.09.42 (4): keepDuplicate (7d)
Wed 2003.10.22_18.03.05 (4): keepMinNumber4, keepWeekDays (60d)
Thu 2003.10.23_08.18.15 (3): keepDuplicate (7d)
Thu 2003.10.23_14.16.24 (3): keepDuplicate (7d)
Thu 2003.10.23_17.00.36 (3): keepMinNumber3, keepWeekDays (60d)
Fri 2003.10.24_13.29.30 (2): keepLastOfWeek (150d), keepMinNumber2, keepWeekDays (60d)
Sun 2003.10.26_09.08.55 (0): keepLastOfMonth (400d), keepLastOfWeek (150d),
                           keepMinNumber1, keepWeekDays (60d)

```

Zusätzlich zu den oben beschriebenen Programmen für das Backup sind noch die Programme `llt` und `multtail` enthalten. `llt` zeigt `creating`, `modify` und `access` time von Dateien an. Mittels `multtail` können mehrere Dateien gleichzeitig wie mit `'tail -f'` verfolgt werden. `Multitail` bietet aber mehr Möglichkeiten als `'tail -f'` und ist wesentlich robuster.

Weitere Pläne

Für die nächsten Versionen von `storeBackup` sind folgende Features geplant:

- Die Hauptzeit beim Erstellen eines Backups wird (außer beim ersten Backup, in dem alles komprimiert / kopiert wird) für die Erzeugung der hard links verbraucht. Das Erzeugen eines hard links ist zwar sehr schnell, durch die relativ zu den anderen Operationen große Anzahl – und die Parallelisierung insbesondere der Kompression – liegt hier jedoch der hauptsächliche Zeitverbrauch. Die nächste Version von `storeBackup` wird die Möglichkeit eröffnen, im ersten Schritt nur die Directory–Struktur und die veränderten Dateien in der Sicherung zu erzeugen. Danach ist das Backup aus Sicht des zu sichernden Mediums abgeschlossen. In einem zweiten Schritt werden dann die noch fehlenden hard links erzeugt. Diese beiden Schritte werden vollständig entkoppelt sein – d.h. sie können auf unterschiedlichen Rechnern laufen und es wird möglich sein, mehrere Backups zu fahren, bevor die neuen hard links erzeugt werden. Nach ersten Messungen wird diese Option einen Performancegewinn gegenüber der "normalen" Komplettsicherung um einen Faktor von 5–10 ergeben, wenn lokal geschrieben wird. Bei einer Sicherung auf einen über NFS gemounteten Rechner wird er deutlich höher ausfallen.
- Für die darauffolgende Version ist geplant, die Möglichkeiten zum Suchen (mit anschließender Rücksicherung) zu erweitern. Es soll möglich sein, in den Sicherungen mit einer beliebigen Regel aus Dateinamen (Pattern), Dateigröße, Erstellungs- / Veränderungszeit, User ID, Group ID, Rechten auf der Datei und einem (einfachen) `grep` auf den Inhalt suchen zu können. Diese Regeln beinhalten dann `'and'`, `'or'`, `'not'` und beliebige Klammerungen.
- Anschließend sehen die künftigen Pläne eine Erweiterung der Optionen (in Anlehnung an `tar`) und die Unterstützung von bisher nicht sicherbaren Dateitypen, wie z.B. Devices vor.

Version und Lizenz

Die zum Zeitpunkt der Erstellung dieses Artikels aktuelle Version des storeBackup Paketes ist 1.14.1. Sie kann, wie schon oben erwähnt, unter <http://www.sourceforge.net/projects/storebackup> heruntergeladen werden.

StoreBackup steht unter der GPL.

<p><u>Webpages maintained by the LinuxFocus Editor team</u> © Heinz-Josef Claes "some rights reserved" see linuxfocus.org/license/ http://www.LinuxFocus.org</p>	<p>Translation information: de --> -- : Heinz-Josef Claes <hjclaes/at/web.de></p>
---	--

2005-01-11, generated by lfparsr_pdf version 2.51