KAPITEL 11

Datensicherung und -wiederherstellung

Jeder Benutzer eines Computersystems weiß, dass Dateien gelegentlich verloren gehen. Ein solcher Verlust kann verschiedene Gründe haben: Benutzer löschen aus Versehen ihre Dateien, ein Bug im Programm zerstört die mühsam eingegebenen Daten, ein Hardwarefehler vernichtet die Daten einer ganzen Festplatte und so weiter und so fort. Der Schaden, der sich aus einem solchen Datenverlust ergibt, kann gering, aber auch sehr teuer sein. Um sich dagegen abzusichern, besteht eine der Hauptaufgaben des Systemverwalters darin, ein Backup-System zu entwerfen und zu implementieren, das periodisch alle Daten des Systems sichert. Es ist ebenfalls Aufgabe des Systemverwalters sicherzustellen, dass die Backups pünktlich und regelmäßig durchgeführt und die Backup-Bänder (und anderen Medien) an einem sicheren und wohl behüteten Platz aufbewahrt werden. Dieses Kapitel beginnt mit einer Beschreibung der Backup-Strategien und -Optionen. Danach wenden wir uns den Werkzeugen zu, die die verschiedenen Unix-Versionen zu diesem Zweck bereitstellen.

Ein ausgezeichnetes Referenzwerk zur Datensicherung auf Unix-Systemen ist *Unix Backup and Recovery* von W. Curtis Preston (O'Reilly & Associates). Es behandelt die hier erläuterten Themen im Detail und geht auch auf Dinge ein, die über den Rahmen dieses Buches hinausgehen (z.B. die Sicherung und Wiederherstellung von Datenbanken).

Pläne für Ernstfälle und alltägliche Anwendungen

Die Entwicklung effektiver Backup-Strategien ist ein fortlaufender Prozess. Üblicherweise erben Sie etwas von Ihrem Vorgänger, wenn Sie ein existierendes System übernehmen, und machen die Dinge, die Sie immer machen, wenn Sie ein neues System bekommen. Das mag für eine Weile gut gehen, aber ich habe Unternehmen gesehen, die versucht haben, an ihren operatorbasierten Backup-Regeln festzuhalten, nachdem ihre mit Mainframes voll gestopften Räume durch dezentrale Workstations, die überall im Gebäude verteilt stehen, ersetzt wurden. Ein solcher Versuch ist ebenso komisch wie heldenhaft, endet aber sehr häufig in Verzweiflung, weil die veralteten Regeln nicht durch neue ersetzt wurden. In einem solchen Moment ist genau der richtige Zeitpunkt zur Entwicklung einer guten Backup-Strategie gekommen, ausgehend davon, wie Sie die Dinge gerade handhaben.

Grundsätzlich stellen Backups eine Art Versicherung dar. Sie repräsentieren Zeit, die Sie aufgewendet haben, um zukünftige Datenverluste zu verhindern. Die Zeit, die für einen Backup-Plan aufgewendet wird, muss mit dem Produktivitätsverlust, dem Termindruck und anderen Dingen verglichen werden, die auftreten, wenn die benötigten Dateien nicht vorhanden sind. Die generelle Forderung, die an jeden Backup-Plan gestellt werden muss, ist, dass er in der Lage sein muss, das gesamte System – oder Gruppen von Systemen – in einer angemessenen Zeitspanne wiederherzustellen, falls es zu einem größeren Fehler kommen sollte. Gleichzeitig sollte die Datensicherung aber komfortabel sein, und die Rücksicherung sollte einfach vonstatten gehen. Die Ansätze, die im Katastrophenfall und bei täglichen Sicherungen verwendet werden, sind häufig sehr verschieden. Der letztendliche Backup-Plan muss aber beide Fälle berücksichtigen (und wird die Unterschiede entsprechend widerspiegeln).

Es gibt sehr viele Faktoren, die bei der Entwicklung eines Backup-Plans zu berücksichtigen sind. Die wohl wichtigsten Fragen sind dabei:

Welche Dateien müssen gesichert werden? Die einfachste Antwort ist natürlich: alle. Obwohl alles außer temporären Dateien und Verzeichnissen irgendwo gesichert werden muss, so muss doch nicht alles als Teil eines System-Backups gesichert werden. Wird das Betriebssystem beispielsweise auf einer CD-ROM ausgeliefert, gibt es eigentlich keinen Grund, die Systemdateien zu sichern, obwohl Sie sich aus Bequemlichkeit doch dafür entscheiden könnten.

Wo sind diese Dateien? Bei dieser Frage geht es sowohl darum, wo die wichtigen Dateien innerhalb des Dateisystems liegen, aber auch darum, welche Systeme die wichtigsten Daten hält.

Wer sichert die Dateien? Die Antwort könnte davon abhängen, wo die Dateien sich befinden. Zum Beispiel liegt die Verantwortung für Server-Backups auf vielen Sites beim Systemadministrator, während gleichzeitig die Benutzer für die Dateien verantwortlich sind, die lokal auf ihren Workstations liegen. Das kann eine gute Idee sein, muss es aber nicht, je nachdem, ob alle wichtigen Dateien wirklich gesichert werden.

Wo, wann und unter welchen Bedingungen sollen Backups durchgeführt werden? Wo steht hier für das Computersystem, auf dem das Backup durchgeführt wird, wobei es sich nicht zwangsläufig auch um das System handeln muss, auf dem sich die Daten befinden. Im Idealfall würden alle Backups außerhalb der normalen Betriebszeiten auf nicht gemounteten Dateisystemen durchgeführt werden. Leider ist das im realen Leben nicht immer möglich.

Wie häufig ändern sich diese Dateien? Diese Information hilft Ihnen bei der Entscheidung, wann und wie oft Backups durchzuführen sind und welche Art von Plan Sie aufstellen müssen. Wenn auf Ihrem System etwa ein großes Entwicklungsprojekt abgewickelt wird, werden sich die Dateien wahrscheinlich häufig ändern und müssen mindestens einmal täglich, vielleicht sogar alle paar Stunden gesichert werden. Wenn Sie andererseits nur eine große Datenbank auf Ihrem System liegen haben, deren Inhalt sich laufend ändert, könnte mehrmals täglich eine Sicherung notwendig

sein, während alle anderen Dateisysteme nur einmal pro Woche gesichert werden müssen.[1]

Wie schnell muss eine fehlende oder beschädigte Datei wiederhergestellt werden? Backups schützen vor dem Verlust vieler, aber auch einzelner Dateien. Der Zeitraum, der nötig ist, um Schlüsseldateien wiederherzustellen, ist ebenfalls eine Sache, die es zu berücksichtigen gilt. Die Anzahl dieser Schlüsseldateien, wie weit verstreut sie im Dateisystem (oder Netzwerk) liegen und wie groß sie sind, sind ebenfalls Faktoren, die in Betracht zu ziehen sind. Vielleicht gibt es bei Ihnen nur eine einzige nicht ersetzbare Datei, aber je nachdem, ob sie 1 KB oder 1 GB groß ist, müssen Sie sehr unterschiedlich planen. (Beachten Sie, dass selbst eine nur 1 KB große Datei für großen Ärger sorgen kann, etwa wenn es sich um die Lizenzdatei handelt, ohne die das Hauptanwendungsprogramm nicht läuft.)

Wie lange müssen die gesicherten Daten behalten werden? Backups schützen aktuelle Daten vor Unfällen. Daher braucht man sie normalerweise nur eine relativ kurze Zeit (ein paar Monate, vielleicht ein bis zwei Jahre), in der sie nützlich sind. Im Gegensatz dazu benötigen die meisten Sites aber auch dauerhafte Archive für wichtige Daten wie beispielsweise die Software und die Daten der Steuererklärung. Diese müssen für längere Zeit gespeichert werden: viele Jahre, vielleicht sogar Jahrzehnte. Zwar sind die Anforderungen ähnlich, aber die Ziele unterscheiden sich deutlich genug voneinander, um normale Backups für Archivierungszwecke als ungeeignet zu erkennen. Über diese Art von Daten und wie man sie erzeugt und archiviert, muss beim Entwurf eines effizienten Backup-Plans ebenfalls nachgedacht werden.

Wo sollen Backup-Medien gelagert werden? Aktuelle Backups werden generell in der Nähe des Computers aufbewahrt, um Daten schnell wiederherstellen zu können. Längerfristige Backups und Archive sollten an einem sicheren, außerhalb gelegenen Ort gelagert werden.

Wohin werden die Daten zurückgeschrieben? Werden die Sicherungsdateien nur auf dem System verwendet, auf dem sie auch angelegt wurden, oder steht zu erwarten, dass sie im Notfall auch auf einem anderen System zum Einsatz kommen? Wenn die Kompatibilität zwischen mehreren Systemen zeitweise oder dauerhaft wichtig ist, muss dies beim Entwurf des Backup- und Recovery-Plans berücksichtigt werden. So müssen Sie beispielsweise sicherstellen, dass jedes auf einem System verfügbare Komprimierungsschema auch auf allen anderen Zielsystemen verarbeitet werden kann (oder vermeiden Sie die Verwendung herstellerspezifischer Formate). Weitere Beispiele für solche Aspekte sind Zugriffskontroll-Listen, die zusammen mit anderen Daten gesichert werden, und das Backup eines Dateisystems von einem Rechner, dessen Dateisystemgröße die des Zielsystems übersteigt.

1 Tatsächlich wird eine Datenbank häufig mit einer vom Hersteller bereitgestellten Software gesichert, aber Sie haben eine Vorstellung von der Aufgabe bekommen.

Backup aktiver Dateisysteme

Nahezu jede Unix-Dokumentation empfiehlt, Dateisysteme (außer dem root-Dateisystem) vor einem Backup mit umount abzukoppeln. Diese Empfehlung wird nur selten befolgt und in der Praxis können Backups auch auf gemounteten Dateisystemen vorgenommen werden. Allerdings müssen Sie die Benutzer darauf aufmerksam machen, dass offene Dateien nicht immer korrekt gesichert werden. Es ist auch richtig, dass es Fälle gibt, bei denen gewisse Ereignisse im aktiven Dateisystem dazu führen können, dass die gesamte Sicherung beschädigt ist. Wir berücksichtigen diejenigen Fälle, die für die verschiedenen Backup-Programme relevant sind, wenn es so weit ist.

Planung der Backup-Kapazitäten

Sobald Sie alle Daten darüber gesammelt haben, was gesichert werden muss und welche Ressourcen dafür zur Verfügung stehen, können Sie eine Prozedur wie die folgende verwenden, um den eigentlichen (detaillierten) Backup-Plan zu entwickeln:

1. Beginnen Sie mit der Festlegung eines idealen Backup-Zeitplans, ohne sich irgendwelchen Zwängen zu unterwerfen, die Ihre aktuelle Situation mit sich bringt. Führen Sie die Daten auf, die gesichert werden sollen, wie häufig diese gesichert werden müssen und welche Unterteilungen der Gesamtdatenmenge sinnvoll sind.
2. Vergleichen Sie den aktuellen Plan nun mit dem, was in Ihrer Umgebung möglich ist. Bedenken Sie dabei auch folgende Punkte:
   - Wenn die Daten zum Backup zur Verfügung stehen: Die Sicherung offener Dateien ist immer problematisch – Sie können bestenfalls auf eine unbeschädigte Momentaufnahme der Datei zum Zeitpunkt des Backups hoffen. Idealerweise werden Datensicherungen also auf ungenutzten Systemen vorgenommen, was üblicherweise außerhalb der normalen Arbeitszeiten geschieht.
   - Wie viele Bandlaufwerke (oder andere Backup-Geräte) stehen zur Verfügung, um die Sicherungen in dieser Zeit durchzuführen? Wie sehen deren maximale Kapazitäten und Übertragungsraten aus? Um Letzteres bestimmen zu können, können Sie mit den Angaben des Herstellers beginnen, sollten aber auch einige eigene Tests unter normalen Bedingungen durchführen, um realistische Transferraten zu ermitteln, die die Systemlasten, Netzwerk-Übertragungsraten und andere Faktoren Ihrer Umgebung berücksichtigen. Sie müssen auch berücksichtigen, ob alle Daten für jedes Backup-Gerät verfügbar sind oder nicht.

   An diesem Punkt führt (wie immer bei Kapazitätsplanungen) kein Weg an der Rechnerei vorbei. Betrachten wir ein einfaches Beispiel: Eine Site mit einem Datenvolumen von 180 GB muss einmal pro Woche gesichert werden und es stehen drei Bandlaufwerke zu Sicherungszwecken zur Verfügung (wir gehen davon aus, dass die Daten allen Laufwerken zugänglich sind). Idealerweise sollten Backups nur wochentags zwischen Mitternacht und sechs Uhr morgens stattfinden. Um das zu schaffen, muss jedes Bandlaufwerk in den 30 Stunden, in denen die Daten zur Verfügung stehen, 60 GB an Daten sichern. Das bedeutet, dass jedes Bandlaufwerk 2 GB pro Stunde (333 KB/sec) auf Band sichern muss.

Das entspricht den Kapazitäten aktueller Bandlaufwerke beim Schreiben lokaler Daten.[2] Andererseits sind viele Daten in unserem Beispiel über das Netzwerk verstreut, d.h., es besteht die Möglichkeit, dass die Daten nicht schnell genug zur Verfügung stehen, um die Höchstgeschwindigkeit der Laufwerke nutzen zu können. Einige Backup-Programme machen auch eine Pause, wenn sie eine offene Datei entdecken, um dem System eine Chance zu geben, die Datei zu schließen (die übliche Wartezeit beträgt 30 Sekunden). Sind viele offene Dateien im Backup-Satz enthalten, kann das die Dauer des Backups deutlich erhöhen.

Darüber hinaus haben wir keinerlei Anstalten getroffen, inkrementelle Backups (die wir später noch erläutern) zwischen vollständigen Backups durchzuführen. Das hier gezeigte Beispiel belastet also die verfügbaren Ressourcen.

3. Nehmen Sie Veränderungen an diesem Plan vor, um die Einschränkungen Ihrer Umgebung zu berücksichtigen. Unsere Beispiel-Site reizt die Möglichkeiten ein wenig zu stark aus, aber es gibt mehrere Möglichkeiten, dies zu ändern:
   - Einbinden zusätzlicher Hardware, in diesem Fall ein viertes Bandlaufwerk.
   - Verringerung der zu sichernden Daten oder der Backup-Häufigkeit. So könnten vollständige Backups für einige Daten zum Beispiel nur alle zwei Wochen durchgeführt werden.
   - Die für Backups verfügbare Zeit erhöhen. (Einige Backups könnten am Wochenende durchgeführt werden und inkrementelle Backups in den frühen Abendstunden.)
   - Backups auf Platte zwischenspeichern. Dieses Schema schreibt die Backup-Archive in einen speziell hierfür vorgesehenen Speicherbereich. Die Dateien können dann zu jeder beliebigen Zeit auf Band geschrieben werden. Festplatten sind außerdem schneller als Bandlaufwerke, so dass diese Methode weniger Zeit beansprucht als das direkte Schreiben auf Band. Natürlich braucht man dazu den nötigen Plattenplatz, um die Archive aufnehmen zu können.
4. Testen und verfeinern Sie den Backup-Plan. Die Praxis offenbart häufig Faktoren, die bei der Planung auf dem Papier übersehen wurden.
5. Überprüfen Sie den Backup-Plan regelmäßig, um sicherzustellen, dass er nach wie vor die beste Lösung für die Backup-Anforderungen Ihrer Site darstellt.

Backup-Strategien

Das einfachste und gründlichste Backup-Schema besteht im Kopieren aller Dateien des Systems auf ein Band. Ein *vollständiges Backup* übernimmt genau diese Aufgabe, wobei alle Dateien innerhalb eines bestimmten Datei-Satzes (häufig die eines einzelnen Computers oder einer einzelnen Partition) berücksichtigt werden.[3]

2 In der Praxis würden Sie natürlich ein Autoloader-Laufwerk benötigen (oder jemanden, der die Bänder mitten in der Nacht wechselt).

3 Für diese Erläuterung beschränke ich mich auf die Betrachtung von partitionsbezogenen Backups. Denken Sie aber daran, dass das nicht der einzige denkbare Weg ist. Ich spreche zwar auch häufig von »Sicherungsbändern«, was ich dabei sage, trifft in den meisten Fällen aber auch auf andere Sicherungsmedien zu.

Vollständige Backups sind zeitaufwendig und können schwer durchführbar sein. Die Wiederherstellung einer einzelnen Datei von einem großen Backup, das sich über mehrere Bänder erstreckt, ist häufig sehr unbequem. Wenn sich die Dateien nicht sehr häufig ändern, steht die Zeit, die für ein vollständiges Backup benötigt wird, in keiner Relation zur Anzahl der tatsächlich neu zu sichernden Dateien. Wenn sich andererseits die Daten sehr schnell ändern und 50 Benutzer nicht arbeiten können, weil eine der Dateien nicht verfügbar ist, oder wenn die Zeit für eine Datensicherung keine Rolle spielt, könnte sogar ein tägliches vollständiges Backup gerechtfertigt sein.

Inkrementelle Backups werden üblicherweise häufiger ausgeführt. Bei einem inkrementellen Backup kopiert das System nur die Dateien, die sich seit dem letzten Backup verändert haben. Dieses Verfahren wird genutzt, wenn vollständige Backups zu lange dauern und sich nur wenige Daten innerhalb eines bestimmten Zeitraums, sagen wir eines Tages, ändern. In solchen Fällen spart man gegenüber den vollständigen Backups eine Menge Zeit.

Einige Unix-Backup-Programme verwenden das Konzept des *Backup-Levels*, um verschiedene Arten von Backups zu unterscheiden. Jeder Backup-Typ besitzt eine eigene Level-Nummer. Per Definition besitzt das vollständige Backup die Level-Nummer 0. Das Backup auf einem bestimmten Level bedeutet, dass alle Dateien gesichert werden, die sich seit dem letzten Backup des nächstkleineren Levels geändert haben. Ein Level-1-Backup sichert also alle Dateien, die sich seit dem letzten vollständigen Backup (Level 0) geändert haben. Ein Level-2-Backup sichert alle Dateien, die sich seit dem letzten Level-1-Backup geändert haben, usw.[4]

Eine typische Backup-Strategie, die mit mehreren Levels arbeitet, führt zu Beginn der Woche ein vollständiges Backup und an den anderen Tagen ein Level-1-Backup durch (sichert also alle Dateien, die sich seit dem letzten vollständigen Backup geändert haben). Der folgende Backup-Plan fasst diese Vorgehensweise zusammen:

Montag: Level 0 (vollständiges Backup)
Dienstag–Freitag: Level 1 (inkrementelles Backup)

Eine Version dieses Ansatzes, die alle sieben Tage berücksichtigt, ist sehr einfach zu konstruieren.

Der primäre Vorteil dieses Plans besteht darin, dass nur zwei Bänder (das vollständige und das inkrementelle Backup) benötigt werden, um das komplette Dateisystem wiederherzustellen. Der Hauptnachteil besteht darin, dass die täglich zu sichernde Datenmenge stetig wachsen wird und – wenn das System sehr aktiv ist – am Ende der Woche die Größe des vollständigen Backups erreichen kann.

Ein weit verbreiteter Monatsplan für Sites mit sehr aktiven Systemen sieht wie folgt aus:

Erster Montag: Level 0 (vollständig)
Alle anderen Montage: Level 1 (wöchentliche inkrementelle Sicherung für Level 0)
Dienstag–Freitag: Level 2 (tägliche inkrementelle Sicherung für Level 1)

4 Nicht alle Backup-Befehle arbeiten explizit mit Level-Nummern. Das Konzept ist aber bei allen verfügbaren Tools gültig oder kann dort implementiert werden, wenn Sie bereit sind, einige der Aufzeichnungen selbst durchzuführen (von Hand oder über ein Skript).

Bei dieser Strategie benötigen Sie drei Bänder für eine komplette Wiederherstellung (die aktuellsten Backups jedes Typs).

Wenn Sie über eine Backup-Strategie nachdenken, müssen Sie auch in Betracht ziehen, wie das System genutzt wird. Die am meisten verwendeten Teile des Dateisystems müssen häufiger gesichert werden als andere (wie etwa das root-Dateisystem, das die Standard-Unix-Programme und -Dateien enthält und sich daher selten ändert). Einige wenige Teile des Systems (etwa */tmp*) müssen niemals gesichert werden. Sie können einige zusätzliche Dateisysteme definieren, die niemals gesichert werden. Jeder, der sie verwendet, ist dann selbst für die Sicherung seiner Dateien verantwortlich.

Sie sollten auch ein vollständiges Backup durchführen, wenn Sie signifikante Änderungen am System, wie etwa die Generierung eines neuen Kernels, das Einbinden eines neuen Anwendungspakets oder die Installation einer neuen Betriebssystemversion, vorgenommen haben. Dabei sollten Sie sich nicht daran orientieren, ob Ihr Sicherungsplan gerade eine vollständige Sicherung vorsieht oder nicht. Es kommt nur gelegentlich vor, dass das root-Dateisystem gesichert wird, wenn Sie aber ein Problem mit Ihrer Systemfestplatte haben, werden Sie es zu schätzen wissen, deutlich weniger Zeit mit der Rekonfiguration verbringen zu müssen.

Unbeaufsichtigte Backups

Der schlimmste Teil beim Anlegen von Backups ist das Herumsitzen und Daraufwarten, dass die Sicherung endlich beendet wird. Auf manchen Sites können unbeaufsichtigte Backups dieses Problem lösen. Passt das Backup auf ein einzelnes Band, können Sie es beim Verlassen des Büros ins Bandlaufwerk legen, den Backup-Befehl in der Nacht automatisch von `cron` ausführen lassen und das Band am nächsten Morgen aus dem Laufwerk nehmen.

Manchmal können unbeaufsichtigte Backups aber ein Sicherheitsrisiko darstellen. Verwenden Sie sie nicht, wenn Unbefugte pysikalischen Zugang zum Bandlaufwerk haben und so das Band stehlen können. Backups müssen genauso geschützt werden wie die wichtigste Datei Ihres Systems.

Sie sollten auch auf unbeaufsichtigte Backups verzichten, wenn Sie nicht verhindern können, dass irgendjemand das Band (oder ein anderes Medium) versehentlich oder absichtlich überschreibt. Das Auswerfen des Bandes nach dem Sicherungslauf kann das manchmal, aber nicht immer verhindern. Wird das Laufwerk häufig benutzt, können Sie diesen Ansatz auch nicht verwenden, weil Ihnen das Laufwerk dann nicht die ganze Nacht zur Verfügung steht.

Verifikation der Daten

In vielen Fällen können die Backups einfach auf das Medium geschrieben werden und das Medium kann direkt am dafür vorgesehenen Ort abgelegt werden. Diese Praxis ist okay, solange Sie hundertprozentig an die Zuverlässigkeit Ihrer Backup-Geräte und -Medien glauben. In allen anderen Fällen ist die Verifikation der Daten eine gute Sache.

Datenverifikation besteht aus einem zweiten Durchlauf der gesicherten Daten, bei dem jede Datei mit der Version auf der Festplatte verglichen wird. Auf diese Weise wird sichergestellt, dass die Datei korrekt gesichert wurde und dass das Medium selbst gelesen werden kann.

Manche Sites werden sich dafür entscheiden, die Daten auf allen Backups zu verifizieren. Alle Sites sollten in regelmäßigen Abständen Verifikationsoperationen für alle Backup-Geräte durchführen. Darüber hinaus fangen viele Geräte mit der Zeit an, »auszuleiern« und Medien zu generieren, die nur im verwendeten Laufwerk erfolgreich gelesen werden können. Wenn Ihre Backups also noch von anderen Geräten oder Systemen gelesen werden müssen, sollten Sie die Lesbarkeit der Medien regelmäßig auch auf den Zielgeräten und -systemen prüfen.

Lagerung der Backups

Die sichere Lagerung der Backup-Bänder, -Disketten oder anderer Medien ist ebenfalls ein wichtiger Teil jedes Backup-Plans. Nachfolgend einige Dinge, die Sie berücksichtigen sollten, wenn Sie entscheiden müssen, wo Ihre Daten gelagert werden sollen:

Wissen, wo die Dinge liegen. Wenn Sie Backups an vorbestimmten Orten lagern, werden Sie sie im Bedarfsfall wesentlich schneller finden. Es ist ebenfalls wichtig, dass jeder, der Daten wiederherstellen muss, weiß, wo sich die Bänder befinden (schließlich werden auch Sie gelegentlich in Urlaub fahren). Installationsmedien, boot-fähige Recover-Bänder, Boot-Disketten und ähnliche Dinge sollten ebenfalls an Stellen gelagert werden, die den Leuten, die sie benötigen könnten, bekannt sind. Ich kann Ihnen aus eigener Erfahrung sagen, dass Systemfehler noch unangenehmer sind, wenn man sich zuerst durch Kisten von Bändern oder CDs wühlen muss, bevor man auch nur eine Chance hat, den Fehler zu beheben.

Ein anderer Aspekt ist zu wissen, welches Band die Datei enthält, die wiederhergestellt werden muss. Die Planung solcher Dinge schließt das Erstellen von Inhaltsverzeichnissen ein, was in diesem Kapitel später noch behandelt wird.

Machen Sie Routine-Restores einfach. Backups sollten nahe genug beim Computer gelagert werden, um verlorene Dateien schnell wiederherstellen zu können. Die Bänder sollten ausreichend gekennzeichnet sein, so dass Sie die benötigten schnell finden.

Idealerweise sollten Sie einen vollständigen Satz von Bändern für jeden Punkt Ihres Backup-Plans besitzen. Wenn Sie zum Beispiel jeden Tag ein Backup anlegen, sollten Sie fünf Bänder besitzen, die Sie jede Woche wiederverwenden können. Wenn Sie sie erübrigen können, sollten Sie sogar 20 haben, so dass Sie die Bänder alle vier Wochen wiederverwenden. Mit nur einem Satz Bändern zu arbeiten beschwört die Probleme geradezu herauf.

Die eindeutige Kennzeichnung der Bänder hilft Ihnen später dabei, die benötigten Daten schnell wiederzufinden. Farbige Aufkleber werden auf vielen Sites als einfache und doch effektive Möglichkeit angesehen, unterschiedliche Sätze von Bändern schnell zu unterscheiden. Ein anderes Extrem wurde auf einer Site verwendet, die ich

einmal besuchte. Dort wurde am Ende jedes Backups ein detaillierter Aufkleber für jedes Band gedruckt.

Versehen Sie Backup-Medien mit einem Schreibschutz. Auf diese Weise verhindern Sie, dass das Backup-Medium versehentlich überschrieben wird. Die zu diesem Zweck verwendeten Mechanismen unterscheiden sich auf den verschiedenen Medien etwas, aber in der Regel geht es darum, eine kleine Scheibe oder ein Schildchen in eine vorbestimmte Position zu bringen. Welche Position die richtige ist, ist unterschiedlich: Bei Disketten, optischen Platten und DAT-Bändern (4 mm) kann auf die Medien geschrieben werden, wenn die Öffnung durch das Schildchen verdeckt wird. Bei 8-mm-Bändern und Wechselplatten kann dagegen nur auf die Medien geschrieben werden, wenn es geöffnet ist.

Umgebungsbedingungen. Die meisten Backup-Medien mögen es kühl, trocken und dunkel. Hohe Luftfeuchtigkeit ist wohl die schädlichste Umgebung, besonders wenn die Medien in Kassetten eingeschlossen sind. Direkte Sonneneinstrahlung sollte ebenfalls vermieden werden, besonders bei Disketten. Die meisten Kunststoffe deformieren sich, wenn sie an einem warmen Sommertag der Sonne direkt ausgesetzt werden. Staub kann für die meisten Backup-Medien ebenfalls zum Problem werden. Beispielsweise konnte ich eine Diskette nicht mehr lesen, nachdem ich sie in meiner Manteltasche mit nach Hause genommen hatte (nun bewahre ich sie immer brav in einem passenden Behälter auf).

Die Tatsache, dass Backup-Medien die gleiche Umgebung bevorzugen, die auch in vielen Computerräumen herrscht, bedeutet nicht notwendigerweise, dass diese Medien auch in dem gleichen Raum aufbewahrt werden sollten. Sie würden so riskieren, dass ein größeres Problem sowohl die Daten des Computers als auch die Sicherungen zerstört. Sicherungsbänder sind für manche Probleme tatsächlich wesentlich anfälliger als Computerkomponenten. Zum Beispiel könnte der Computer bei einem Rohrbruch nur leichten Schaden nehmen, während die Sicherungsbänder durch die Feuchtigkeit völlig zerstört werden.

Wenn sich der Ort, an dem die Sicherungsbänder gelagert werden, bezüglich der Temperatur um mehr als ein paar Grad vom Computerbereich unterscheidet, sollten sich die Bänder an die Umgebungstemperatur des Computers akklimatisieren, bevor man etwas darauf schreibt.

Magnetfelder sind ebenfalls zu berücksichtigen. Einer der technischen Korrektoren des Buches gab eine Geschichte zum Besten über »eine ganze Backup-Bibliothek, die nahezu täglich zerstört wurde«. Wie sich herausstellte, wurden die Bänder zwar an einem sicheren Ort gelagert, standen aber an einer Wand, auf deren Rückseite sich ein Lastenaufzug befand. Die durch diesen Aufzug verursachten magnetischen Felder sorgten dafür, dass all die schönen Sicherungsbänder gelöscht wurden. Lustig, aber belehrend.

Behandeln Sie die Medien richtig. Manche Medien stellen spezielle Anforderungen, die Sie berücksichtigen müssen. Zum Beispiel sollten Disketten und Zip-Medien auf der schmalen Kante stehend und nicht übereinander gestapelt aufbewahrt werden. Auch

Cartridges möchten mit vertikal ausgerichteten Spulen (wie die Reifen eines Autos) gelagert werden, und zwar mit der die Köpfe berührenden Seite nach unten. Wenn Sie darauf zählen, dass sie wichtige Daten für Sie aufbewahren, sollten Sie sie gut behandeln und sie so lagern, wie sie es verlangen.

Denken Sie an die Sicherheit. An jedem Ort, an dem Sicherungsbänder aufbewahrt werden, müssen die Regeln physikalischer Sicherheit gelten: Die Bänder sollten so weit wie möglich vor Diebstahl, Vandalismus und umgebungsbedingten Katastrophen geschützt sein.

Lagerung über lange Zeiträume und an anderen Orten

An anderen Orten gelagerte Backups sind die letzte Barriere zwischen Ihrem System und der totalen Vernichtung. Diese vollständigen Backups werden an einem abschließbaren, feuerfesten Ort aufbewahrt, dessen Umgebungsbedingungen vollständig überwacht werden und der sich an einem völlig anderen Ort befindet als Ihre Site. Falls irgend möglich, sollten solche Backups an nicht gemounteten Dateisystemen vorgenommen werden.

Die Vorbereitung eines Backups, das an einem anderen Ort gelagert werden soll, ist einer der wenigen Anlässe, bei denen das einfache Anlegen eines Backups nicht genügt.[5] In diesen Fällen müssen Sie auch sicherstellen, dass die Sicherungsmedien wirklich gelesen werden können. Führen Sie dazu den Restore-Befehl aus, der den Inhalt des Mediums ausgibt. Zwar haben Sie auf diese Weise immer noch keine Garantie, dass die Dateien alle gelesen werden können, aber die Wahrscheinlichkeit ist doch recht hoch. Manche Backup-Utilities stellen entsprechende Prüfmechanismen bereit, bei denen alle Dateien des Backups mit denen auf der Festplatte verglichen werden. Diese Methode sollte, wenn möglich, bei allen kritischen Backups verwendet werden, leider sind diese Prüfmechanismen selten. Diese Methode ist für die Prüfung kritischer Backups zu bevorzugen. Wann immer die Integrität des Backups von Bedeutung ist, sollte der beste verfügbare Prüfmechanismus verwendet werden.

Permanente Backups

Bei für die permanente Archivierung vorgesehenen Daten sollten zwei Backup-Sätze angelegt werden. Die einfache Idee dahinter ist die, dass die redundante Kopie genutzt werden kann, wenn die erste fehlerhaft ist. Das Medium sollte regelmäßig geprüft werden (jährlich, möglichst halbjährlich). Fällt ein Medium aus – und das machen irgendwann alle – , müssen Sie vom zweiten Medium eine Kopie anlegen und die defekte ersetzen.

Sie sollten auch sicherstellen, dass mindestens ein funktionierendes Laufwerk des Typs zur Verfügung steht, mit dem Sie permanente Backups anlegen. Wenn Sie zum Beispiel ein Archiv mit 8mm-Bändern besitzen, brauchen Sie auch ein funktionierendes 8mm-Laufwerk, um diese Bänder lesen zu können. Das gilt so lange, bis sich Ihr primäres Backup-Medium ändert. Gleiches gilt natürlich für die Pflege der Softwarepakete und der Laufzeitumgebung, die notwendig sind, um die Daten auch wirklich nutzen zu können.

5 Ein anderer solcher Zeitpunkt ist der Wiederaufbau des Dateisystems.

Schließlich sollten Bänder regelmäßig (vielleicht zweimal im Jahr) vor- und zurückgespult werden, um ihre Lesbarkeit zu erhalten. Auf Grund dieser Anforderungen sind, was die permanente Speicherung von Daten angeht, Bänder durch CDs ersetzt worden.

Wann Zwanghaftigkeit gut ist

Es ist sehr leicht, auf Backups zu verzichten, besonders wenn Sie nur für Ihre eigenen Dateien verantwortlich sind. Die regelmäßige Durchführung von Datensicherungen ist aber lebenswichtig. Grundsätzlich ist es eine gute Idee, sich vorzustellen, dass Sie sich an Ihren Rechner setzen und feststellen, dass alle Festplatten einen Headcrash hatten. Mit einer solchen Katastrophe im Hinterkopf wird deutlich, wozu und wie häufig Datensicherungsläufe notwendig sind. Backups sind nützlich, wenn Dateien versehentlich gelöscht wurden, aber sie sind von elementarer Bedeutung, wenn es zu ernsten Hardwarefehlern oder anderen Katastrophen kommt, und diese Katastrophen werden passieren. Jede Hardware besitzt nur eine beschränkte Lebenserwartung, und irgendwann geht einfach etwas schief.

Wenn wir dieser Tatsache ins Auge sehen, wird ganz klar, warum ein zwanghaftes Festhalten an der Routine für einen effektiven Systemadministrator eine so wichtige Eigenschaft ist. Sich die schlimmsten Fälle vorzustellen ist ein Teil seiner Arbeit. Sollen die anderen Sie ruhig zwanghaft nennen, eines Tages wird Ihre Zwanghaftigkeit sie retten, oder zumindest ihre Daten.

Backup-Medien

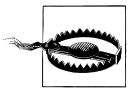
Als ich mit meiner Arbeit als Systemadministrator begann, waren 9-Spur-Bänder das einzige Medium, das für Backups in Frage kam.[6] Dies trifft nicht länger zu; heute gibt es eine Vielzahl verschiedener Medien, die für die Speicherung von Backups geeignet sind. Dieser Abschnitt bietet eine kurze Übersicht über die verfügbaren Medien. Diese Liste enthält die meisten heute gängigen Laufwerks- und Medientypen. Die Backup-Strategie für ein bestimmtes System umfasst häufig mehr als einen Medientyp.

Magnetband

Die eine oder andere Art von Magnetband war für Jahrzehnte das traditionelle Backup-Medium. Mit der Zeit kam es zu einer Vielzahl von Größen und Formen. Angefangen hat das Ganze mit den 7- und 9-Spur-Bändern: 0,5 Zoll breiten Bändern, die auf einer Spule aufgewickelt waren. Die Einführung einer das Band und die Spulen enthaltenden Kassette war ein wesentlicher Schritt bei der Reduzierung der Platzanforderungen von Backup-Medien. Das erste Band dieser Art war die 1/4-Zoll-Cartridge (auch als QIC-Tape bekannt), die eine ganze Zeit lang *das* Medium für die meisten Workstations war. Gelegentlich werden diese Bänder heute immer noch verwendet.

6 Die einzigen anderen Möglichkeiten waren Lochkarten und Lochstreifenpapier.

Vor etwa 20 Jahren standen dann auch Bänder mit höheren Kapazitäten zur Verfügung, deren Formate ursprünglich für andere Märkte entwickelt worden waren. 8-mm-Bandlaufwerke wurden in den späten 80ern populär und sind immer noch weit verbreitet. Ursprünglich für Video-Zwecke entwickelt, weisen die Bänder in etwa die Größe einer Audiokassette auf. 4-mm-Digital-Audio-Bänder (üblicherweise als DAT-Bänder bezeichnet, auch wenn das zur Datenspeicherung verwendete Schema technisch als DDS bekannt ist) sind ebenfalls weit verbreitet. DAT-Bänder sind etwa 25% kleiner als 8-mm-Bänder.



8-mm- und 4-mm-Bänder gibt es in zwei Güteklassen. Die einen sind für Video- bzw. Audioaufnahmen gedacht und die besseren (und teureren) eignen sich zur Speicherung von Daten. Achten Sie darauf, nur die zur Datenspeicherung gedachten Bänder zu verwenden. Auch wenn die Bänder mit der niedrigeren Qualität scheinbar auch funktionieren, sind sie meiner Erfahrung nach (und allen bekannten Legenden zum Trotz) deutlich unzuverlässiger.

Beide Bandtypen werden heute verwendet, auch wenn DAT wesentlich verbreiteter ist als 8 mm. Beide Bandtypen gibt es in verschiedenen Längen und entsprechenden Datenkapazitäten. Momentan sind die längsten normalen 8-mm-Bänder 160 Meter lang und können bis zu 7 GB an Daten aufnehmen.[7] Es gibt aber auch Bänder für 1,2 GB (54 m) und 2,4 GB (112 m). DAT-Bänder entsprechen den verschiedenen DDS-Leveln:

DDS-1
: 2- und 3-GB-Bänder (60 und 90 m)

DDS-2
: 4-GB-Bänder (120 m)

DDS-3
: 12-GB-Bänder (125 m)

DDS-4
: 20-GB-Bänder (150 m)

DDS-3 und DDS-4 verwenden eine andere Technik als die früheren Versionen.

Bedenken Sie, dass nur die neuesten Bandlaufwerke die längsten Bänder unterstützen, aber die meisten Laufwerke sind rückwärts kompatibel (zumindest was das Lesen der Bänder angeht).

Es gibt auch verschiedene neuere Magnetband-Technologien. Mammoth-2[8] von Exabyte und AIT (Advanced Intelligent Tape) von Sony verhelfen 8-mm-Bändern zu wesentlich höheren Speicherkapazitäten: 20, 40, oder 60 GB bzw. 35 oder 50 GB. Beide verwenden die von Sony entwickelte AME-Kassette (Advanced Metal Evaporative, ein neues 8-mm-Format). Einige Mammoth-2-Laufwerke können auch alte 8-mm-Bänder lesen, verlangen

7 Das heißt 7 GB an Bits.

8 Die dem vorausgegangene Mammoth-Technologie war für ihre Unzuverlässigkeit berüchtigt. Mammoth-2 scheint besser zu funktionieren.

allerdings nach jeder Instanz eine umfangreiche Säuberung. Diese Laufwerke zählen mit Transferraten von bis zu 12 MB/s bei Mammoth-2 und 6 MB/s bei AIT-Laufwerken zu den schnellsten.

Die DLT-Technologie (Digital Linear Tape) wurde ursprünglich von der Digital Equipment Corporation entwickelt, wurde aber später an die Quantum Corporation verkauft. Dieses Format verwendet Cartridges, die an DECs alte TK-Familie erinnern, die sich als extrem zuverlässig und langlebig erwiesen hat. Dieses Format ist mit Transferraten von bis zu 10 MB/s ebenfalls recht schnell.

Ihre hohe Kapazität macht die Magnetbänder zum idealen Medium für unbeaufsichtigte Backups: Sie können abends ein Band einlegen, das Skript starten, das verschiedene Dateisysteme auf Band sichert, und dann nach Hause gehen.

Bänder haben aber auch einige Nachteile:

- Sie sind extrem anfällig gegen Hitze und Magnetfelder und recht nachtragend, wenn man sie nicht ordentlich behandelt. Elektromagnetische Felder werden von einer ganzen Reihe gängiger Geräte erzeugt, die man in der Nähe von Computern findet. Hierzu gehören unterbrechungsfreie Stromversorgung, externe Peripheriegeräte mit eigener Stromversorgung, Monitore und Lautsprecher. Alleine das einfache Lesen eines Magnetbands sorgt schon für eine Abnutzung.
- Es handelt sich um sequenzielle Speichergeräte. Um eine bestimmte Datei auf dem Band zu erreichen, müssen Sie das Band an die richtige Stelle spulen. Das ist eher ein Problem alter Bandlaufwerke; aktuelle High-End-Laufwerke erreichen einen beliebigen Punkt des Bandes in Sekundenschnelle.

Magneto-optische Platten

Magneto-optische Platten haben die gleiche Größe wie Disketten, sind aber doppelt so dick und können wesentlich mehr Daten speichern. Magneto-optische Platten gibt es in Versionen zu 3,5 und 5,25 Zoll.[9] Die Kapazitäten erreichen bis zu 9,1 GB. Optische Platten sind wesentlich stabiler als rein magnetische Medien. Diese Stabilität verdanken sie der Tatsache, dass die Daten magnetisch geschrieben, aber optisch gelesen werden. Das Lesen der Platte hat also keinen negativen Effekt auf die gespeicherten Daten. Darüber hinaus lässt sich das Medium ganz nach Bedarf löschen und wieder beschreiben. Schließlich haben magneto-optische Platten auch den Vorteil des wahlfreien Zugriffs. Die Transferraten dieser Geräte erreichen in etwa 5 MB/s.

Aktuelle Laufwerke sind noch recht teuer – über 2000 Dollar –, ebenso wie die Medien selbst, aber trotzdem sehr populär. In der vorherigen Auflage dieses Buches (etwa 1995) habe ich geschrieben, es sei wohl anzunehmen, »dass ein wieder beschreibbares Medium,

9 Vielleicht wundern Sie sich, was an 3,5 Zoll und 5,25 Zoll so toll ist. Es ist die Tatsache, dass Geräte dieser Größe in die Geräteeinschübe von PCs und die Medien in die entsprechenden Ablageboxen passen.

das permanent über ein Gigabyte an Daten auf der Größe einer Diskette speichern kann, eine Zukunft hat«. Nun sehen wir mehrere Gigabyte an Daten und definitiv eine Zukunft.

Einige weitere optische Formate werden von anderen Herstellern verwendet oder entwickelt.

CDs und DVDs

Schreibbare CDs und DVDs sind auf Grund drastischer Preisreduzierungen sowohl bei den Laufwerken als auch bei den Medien zu erschwinglichen Backup-Medien geworden. Es gibt zwei Arten schreibbarer CDs, die als CD-R (Write-Once CD) und CD-RW (Rewriteable, d.h. wiederbeschreibbare CDs) bezeichnet werden. Beide besitzen eine Kapazität von 640 MB, mittlerweile sind aber auch 700 MB CD-Rs verfügbar.

Schreibbare DVDs erschließen sich zu dem Zeitpunkt, als diese Zeilen geschrieben werden, gerade den Massenmarkt. Es gibt verschiedene DVD-Aufzeichnungsformate:

DVD-RAM
: Das erste verfügbare Format. Wird mittlerweile immer weniger eingesetzt, weil es mit normalen DVD-Laufwerken nicht gelesen werden kann.

DVD-R
: Write-Once-DVDs (ebenfalls eine alternde Technologie).

DVD-RW
: Wiederbeschreibbare DVDs, die von normalen DVD-Laufwerken gelesen werden können.

DVD+RW
: Eine gerade eingeführte Technologie, die von mehreren Laufwerksherstellern entwickelt wurde. Diese Laufwerke erzeugen normale (sequenzielle) DVDs ebenso wie Platten mit wahlfreiem Zugriff. Erstere können von normalen DVD-Playern (aber nicht von Recordern anderer Typen) gelesen werden, auch wenn einige ältere Modelle möglicherweise ein Firmware-Update benötigen. DVD+RW-Medien können bis zu 4,7 GB pro Seite aufnehmen.

  Während dies geschrieben wird, hat Hewlett-Packard gerade einen sehr preisgünstigen DVD+RW-Writer auf den Markt gebracht, der für PC-basierte Systeme geeignet ist. Dieses System könnte sich mit der Zeit auf diesem Sektor zu einem populären Backup-Gerät mausern.

Wechselplatten: Zip und Jaz

Wechselplatten sind vollständig abgeschlossene Platteneinheiten, die ganz nach Bedarf in ein Laufwerk eingelegt werden. Sie sind deutlich zuverlässiger als Bänder oder Disketten. Auf Unix-Systemen verhalten sie sich im Allgemeinen wie Festplatten, man kann sie aber auch als riesige Diskette betrachten. Sie sind in einigen Umgebungen und unter bestimmten Umständen als Backup-Medien geeignet.

Über die Jahre hat es eine ganze Reihe von Wechselplatten-Technologien gegeben. Die Zip- und Jaz-Laufwerke von Iomega dominieren diesen Markt mittlerweile. Zip-Laufwerke – mit Größen zwischen 100 MB und 250 MB – können auf den meisten Unix-Systemen eingesetzt werden. Jaz-Laufwerke mit einer Kapazität von 1 GB oder 2 GB können ebenfalls verwendet werden. Ich hatte große Schwierigkeiten mit den frühen Jaz-Laufwerken, die für eher seltene, unregelmäßige Backups konzipiert waren und ständig Probleme hatten, selbst wenn man sie nicht fortlaufend nutzte. Neuere Laufwerke sollen da besser sein. Beide Laufwerkstypen sind mit verschiedenen E/A-Schnittstellen verfügbar: SCSI, USB, IDE.

Floppy-Disks

Floppy-Disks finden sich immer noch auf den meisten PC-basierten Computersystemen[10] und haben durchaus einen (wenn auch geringen) Nutzen für die Datensicherung. Zum Beispiel verwenden PC-basierte Unix-Versionen (ebenso wie einige auf größeren Systemen laufende) Floppy-Disks als Notfall-Boot-Geräte. Zusätzlich können Floppy-Disks für Backup-Aufgaben wie etwa das Speichern angepasster System-Konfigurationsdateien des root-Dateisystems sehr nützlich sein. Standarddisketten können 1,44 MB aufnehmen und einige Unix-Workstations besitzen Laufwerke, die diese Kapazität auf 2,8 MB verdoppeln. Gelegentlich könnten Ihnen auch so genannte Super-Disks unterstützende Disketten-Laufwerke begegnen: wie normale Disketten aussehende Medien, die aber 120 MB aufnehmen können.

Festplatten

Es ist fast immer möglich, Daten eines Dateisystems auf einer anderen Festplatte zu sichern. In manchen Situationen kann dies eine durchaus gute Lösung sein. Die bei vielen Computern und RAID-Systemen integrierten Möglichkeiten der Datenspiegelung binden eine solche Art des Backups schon auf Dateisystemebene ein.

Angesichts der heutzutage sehr niedrigen Preise für Festplatten können sie in manchen Situationen eine recht brauchbare Backup-Lösung darstellen. Zum Beispiel stellen einige Sites im lokalen Netzwerk eine große Backup-Festplatte zur Verfügung, auf der Benutzer regelmäßig Kopien ihrer Schlüsseldateien ablegen können. Große Festplatten können auch zu Hilfszwecken verwendet werden, als temporäre Datenspeicher, als Bereiche, um Daten vorzuhalten, und ähnliche Dinge. Sie können auch als Zwischenspeicher verwendet werden, in dem Backups temporär abgelegt werden, bevor sie auf Band oder andere Medien geschrieben werden.

Stacker, Jukeboxen und ähnliche Geräte

Es existieren eine ganze Reihe von Geräten, die die Verwendung von Medien weiter automatisieren sollen und eine große Zahl von Medien speichern und verfügbar machen sollen. Zum Beispiel gibt es automatisch ladende Bandlaufwerke – auch als *Stacker* oder *Stackloader-Laufwerke* bekannt – , die Bänder automatisch aus einem Stapel (von etwa 10 Bän-

10 Auch wenn das in ein paar Jahren wahrscheinlich nicht mehr zutrifft.

dern) auswählen und einlegen. Frühe Stacker konnten nur nacheinander auf die Bänder zugreifen, viele aktuelle Geräte können auf jedes beliebige Band zugreifen.

Andere Geräte enthalten mehrere Laufwerke in einer Box, die sich dem Benutzer als ein einziges Bandlaufwerk präsentiert und die Kapazität all ihrer Komponenten bereitstellt. Alternativ können solche Geräte mehrere identische Kopien gleichzeitig anlegen.

Wieder andere Geräte kombinieren mehrere Laufwerke und Autoloading-Fähigkeiten. Solche Geräte werden auch als *Jukeboxen* oder *Libraries* bezeichnet.[11] Die besten Geräte sind in der Lage, ein bestimmtes Band herauszusuchen und ins gewünschte Laufwerk einzulegen. Einige besitzen integrierte Barcode-Leser, so dass die Bänder über ihr Label und nicht über ihre physikalische Lage identifiziert werden können. Ähnliche Geräte existieren auch für optische Laufwerke und wiederbeschreibbare CD-ROMs.

Lebenserwartung von Medien

Von Zeit zu Zeit müssen Sie auch über die realistisch zu erwartende Lebenserwartung Ihres Backup-Mediums nachdenken. Unter den richtigen Bedingungen gelagert, können Bänder mehrere Jahre halten, aber unglücklicherweise können Sie sich darauf nicht verlassen. Einige Hersteller empfehlen den Austausch der Bänder einmal pro Jahr. Wenn Sie sich das leisten können, ist das auch eine gute Idee. Die Art und Weise, in der Bänder und Disketten gelagert werden, können deren Lebenserwartung beeinflussen: Sonnenlicht, Hitze und Feuchtigkeit können sie deutlich reduzieren. Ich ersetze jedes Band, bei dem ein Lesefehler oder ein anderer Fehler mehr als einmal aufgetreten ist, und zwar unabhängig von seinem Alter. Für einige Menschen und Fälle reicht schon ein einzelner Fehler aus. Floppy- und Zip-Disketten werfe ich beim ersten Anzeichen eines Problems weg.

Auch so genannte permanente Medien wie CDs haben eine beschränkte Lebenserwartung. Zum Beispiel fangen CDs nach ungefähr fünf Jahren (und manchmal früher) an, fehlerhaft zu werden. Dementsprechend ist das Anlegen von Kopien wichtiger Daten und deren regelmäßige Überprüfung die einzig sinnvolle Vorgehensweise.

Nach diesen Überlegungen sollten Sie für Ihre Site alternative Medien für außerbetriebliche und der Archivierung dienende Backups erwägen. Die Hersteller optischer Platten geben zum Beispiel eine Lebenserwartung von 15 Jahren für ihre Medien an. (Diese Angaben basieren auf beschleunigten Alterungstests. Während wir diese Zeilen schreiben, wissen wir aber erst in etwa 8-9 Jahren, ob diese Behauptung wirklich stimmt.)

Vergleich von Backup-Medien

Tabelle 11-1 führt die wichtigsten Eigenschaften einer Vielzahl unterschiedlicher Backup-Medien auf. Die Medienkapazitäten waren zum Zeitpunkt, als diese Zeilen geschrieben

11 Sehr große Libraries (mit mehr als 500 Volumes) werden auch *Silos* genannt. Die zwei Arten von Geräten wurden danach unterschieden, ob mehrere Hosts angeschlossen werden konnten oder nicht, aber einige Libraries besitzen diese Fähigkeit mittlerweile auch. Separate Silos sind auch in der Lage, Bänder untereinander auszutauschen.

wurden, die größten, die zur Verfügung standen. Die Größenangaben beziehen sich auf die reine Datenkapazität: die tatsächliche Menge an Daten, die auf das Medium geschrieben werden kann.

Der Laufwerkspreis ist der günstigste momentan verfügbare Preis und geht von der günstigsten E/A-Schnittstelle aus. SCSI-Versionen vieler Geräte, die auch als IDE-Einheiten verfügbar sind, kosten mindestens 15% mehr (manchmal sogar noch mehr). Mit 100 Dollar kostet ein USB-Diskettenlaufwerk 10-mal mehr als ein normales.

Die Preise für die Medien beziehen sich auf die niedrigsten allgemein verfügbaren Preise bei Abnahme größerer Mengen (etwa 50–100 für CDs) mit einfacher Verpackung (bei CDs zum Beispiel eine Spindel und keine Jewel-Cases). Alle Preise wurden Mitte 2002 in den Vereinigten Staaten ermittelt.

Die Spalte mit der minimalen Lebenserwartung gibt einen ungefähren Wert für die Lebenserwartung des genannten Mediums an. Natürlich können einzelne Medien in manchen Fällen auch früher defekt sein.

Tabelle 11-1: Populäre Backup-Geräte und -Medien

Typ	Kapazität	Preis Gerät[a]	Preis Medium[a]	Minimale Lebenserwartung
Diskette	1,44 MB[b]	$10	$0,25	2 Jahre
Super-Disk	120 MB	$120	$8	2–3 Jahre
Zip-Laufwerk	100 MB	$70	$5	3–5 Jahre
	250 MB	$140	$12	3–5 Jahre
Jaz-Laufwerk	1 GB	$300	$80	4–5 Jahre
	2 GB	$340	$100	4–5 Jahre
CD-R	700 MB (80 Minuten)	$150	$0,85	5 Jahre
CD-RW	640 MB (74 Minuten)	$150	$1	5 Jahre
DVD-R	4,7 GB (eine Seite)	$700	$8	5 Jahre?
	9,4 GB (beidseitig)	$700	$40	5 Jahre?
DVD+RW	4,7 GB	$600	$8	5 Jahre?
DAT-Band, 4 mm DDS	4 GB (120 m DDS-2)	$550	$6	3–4 Jahre
	12 GB (125 m DDS-3)	$700	$12,50	3–4 Jahre
	20 GB (150 m DDS-4)	$1200	$26	3–4 Jahre
8-mm-Band	7 GB (160 m)	$1200	$6	2–4 Jahre
Mammoth-2 (AME)	20 GB	$2500	$36	3–4 Jahre?
	60 GB	$3700	$45	3–4 Jahre?
AIT-Band	35 GB	$900	$79	3–4 Jahre?

Tabelle 11-1: Populäre Backup-Geräte und -Medien (Fortsetzung)

Typ	Kapazität	Preis Gerät[a]	Preis Medium[a]	Minimale Lebenserwartung
	50 GB	$2600	$85	3–4 Jahre?
	100 GB	$3900	$105	3–4 Jahre?
DLT	40 GB	$4000	$70	10 Jahre
SuperDLT	110 GB	$6000	$150	10 Jahre
Magneto-optisch (RW)	5,2 GB	$2300	$65	15 Jahre?
	9,1 GB	$2700	$93	15 Jahre?
Festplatte	100 GB (IDE)	k. A.	$2–3/GB	5–7 Jahre
	180 GB (SCSI)	k. A.	$10/GB	5–7 Jahre

a Ungefährer minimaler Preis in US-Dollar.
b Einige wenige von Unix-Herstellern angebotene Diskettenlaufwerke erhöhen die maximale Kapazität auf 2,8 MB.

Gerätedateien für Bandlaufwerke

Traditionell haben die Gerätedateien für Bandlaufwerke Namen der Form */dev/rmt*n oder */dev/rmt/*n, wobei *n* für die Laufwerksnummer steht. Der Zugriff auf Bandlaufwerke erfolgt nahezu immer über die zeichenorientierte Gerätedatei (Raw-Device). Momentan enthalten die Namen der Gerätedateien üblicherweise weitere Zeichen als Präfixe und/oder Suffixe. Diese zusätzlichen Zeichen geben an, wie auf das Gerät zuzugreifen ist: die zu verwendende Dichte (density), ob die in das Laufwerk integrierte Hardware-Komprimierung verwendet werden soll, ob das Band nach der Operation zurückgespult werden soll und so weiter.

AIX-Systeme besitzen außerdem noch Suffixe, mit denen Sie festlegen können, ob das Band vor der Verwendung einem so genannten Retensioning unterzogen werden soll. Retensioning bezeichnet die gleichmäßige Verteilung der Spannung eines Bandes, indem man das Band zuerst an den Anfang spult, dann an das Ende und dann wieder zurück an den Anfang. Das ist sogar noch langsamer, als es sich anhört. Die Idee ist, latent lockere Bereiche zu beseitigen. In der Praxis ist das allerdings selten notwendig.

Tabelle 11-2 führt die aktuellen Namenskonventionen für die Gerätedateien der von uns betrachteten Betriebssysteme auf.

Tabelle 11-2: Namen der Gerätedateien für Bandlaufwerke

Unix-Version	Format und Beispiele[a]	Präfixe/Suffixe	Manpage
AIX	*/dev/rmt*n[.m] */dev/rmt0.1* */dev/rmt0.5* **Hinweis**: Die Komprimierung wird mit dem `chdev`-Befehl aktiviert und deaktiviert.	m: ohne=Zurückspulen, kein Retensioning, Low Density *1*=kein Zurückspulen, kein Retensioning, Low Density *2*=Zurückspulen, Retensioning, Low Density *3*=kein Zurückspulen, Retensioning, Low Density *4*=Zurückspulen, kein Retensioning, High Density *5*=kein Zurückspulen, kein Retensioning, High Density *6*=Zurückspulen, Retensioning, High Density *7*=kein Zurückspulen, Retensioning, High Density	`rmt(4)`
FreeBSD	*/dev/*[*n*]*rast*n */dev/*[*e*|*n*]*rsa*n */dev/nrast0* */dev/nrsa0*	*n*=kein Zurückspulen *e*=Band nach Verarbeitung auswerfen (Density und Komprimierung werden über das `mt`-Utility gewählt.)	`ast` `sa(4)`
HP-UX	*/dev/rmt/*c*i*t*j*d*0*TYP[*b*][*n*] */dev/c0t3d0DDSbn* */dev/c0t3d0BESTbn*	i=Controller j=SCSI ID *n*=kein Zurückspulen *b*=BSD-artige Fehlerkontrolle verwenden TYP=Bandtyp und/oder Dichte angebendes Schlüsselwort (z.B. *BEST*, *DDS*)	`mt(7)`
Linux	*/dev/*[*n*]*st*nx */dev/nst0* */dev/nst0m*	*n*=kein Zurückspulen x: ohne=Standarddichte *l*=Low Density *m*=Medium Density *a*=Dichte automatisch wählen	`st`
Solaris	*/dev/rmt/*nx[*b*][*n*] */dev/rmt/0lbn* */dev/rmt/0hbn*	*b*=BSD-artige Fehlerkontrolle verwenden *n*=kein Zurückspulen x: ohne=Standarddichte *l*=Low Density *m*=Medium Density *h*=High Density *c*=Hardware-Komprimierung verwenden	`st`

Tabelle 11-2: Namen der Gerätedateien für Bandlaufwerke (Fortsetzung)

Unix-Version	Format und Beispiele[a]	Präfixe/Suffixe	Manpage
Tru64[b]	*/dev/*[*n*]*rmt/tape*n_*d*m	m:	`tz`
	/dev/nrmt/tape0_d2 */dev/nrmt/tape0_d3*	*0*=Low Density, Komprimierung *1*=High Density, Komprimierung *2*=Low Density, Komprimierung *3*=High Density, Komprimierung (die Werte 4–7 sind für einige Laufwerke ebenfalls definiert)	

a In allen Fällen steht n für die Laufwerksnummer. Die Beispiele sind alle für nicht zurückspulende Bandlaufwerke mit ausgeschalteter Hardware-Komprimierung mit niedrigster und höchster Dichte (nach Verfügbarkeit).
b Ältere Tru64-Systeme verwenden nunmehr veraltete Gerätenamen der Form */dev/tz** und */dev/ta**.

Einige Systeme stellen einfachere Namen als Links auf häufig verwendete Bandgeräte zur Verfügung. Sie können herausfinden, auf welches Gerät diese Namen verweisen, indem Sie sich ein langes Verzeichnis-Listing ansehen. Hier ein Beispiel aus einem HP-UX-System:

```
crw-rw-rw- 2 bin bin 205 0x003000 Oct 7 1999 0m
crw-rw-rw-  2 bin    bin    205 0x003080 Oct  7  1999 0mb
crw-rw-rw-  2 bin    bin    205 0x003040 Oct  7  1999 0mn
crw-rw-rw-  2 bin    bin    205 0x0030c0 Oct  7  1999 0mnb
crw-rw-rw-  2 bin    bin    205 0x003000 Oct  7  1999 c0t3d0BEST
crw-rw-rw-  2 bin    bin    205 0x003080 Oct  7  1999 c0t3d0BESTb
crw-rw-rw-  2 bin    bin    205 0x003040 Oct  7  1999 c0t3d0BESTn
crw-rw-rw-  2 bin    bin    205 0x0030c0 Oct  7  1999 c0t3d0BESTnb
crw-rw-rw-  1 bin    bin    205 0x003001 Oct  7  1999 c0t3d0DDS
crw-rw-rw-  1 bin    bin    205 0x003081 Oct  7  1999 c0t3d0DDSb
crw-rw-rw-  1 bin    bin    205 0x003041 Oct  7  1999 c0t3d0DDSn
crw-rw-rw-  1 bin    bin    205 0x0030c1 Oct  7  1999 c0t3d0DDSnb
```

In diesem Fall verweisen *0m* und *c0t3d0BEST* auf das gleiche Bandlaufwerk und den gleichen Zugriffsmodus (genau wie die entsprechenden mit Suffixen versehenen Formen).

Das Standard-Bandlaufwerk eines Systems ist üblicherweise das erste Laufwerk in seinem Standardmodus (zurückspulend):

AIX	*/dev/rmt0*
FreeBSD	*/dev/rsa0*
HP-UX	*/dev/rmt/0m*
Linux	*/dev/st0*
Solaris	*/dev/rmt/0*
Tru64	*/dev/rmt/tape0_d0*

Auf Linux-Systemen (und einigen anderen) ist das Gerät */dev/tape* ein Link auf das Standard-Bandgerät des Systems. Sie können diesen Link auf jedes beliebige Laufwerk zeigen lassen, indem Sie den Link neu anlegen. Auf FreeBSD-Systemen verwenden einige Befehle die Umgebungsvariable *TAPE*, um das Standard-Bandlaufwerk zu bestimmen.

Bandgeräte-Attribute bei AIX

Auf AIX-Systemen können Sie den `lsattr`-Befehl verwenden, um sich die Attribute eines Bandlaufwerks anzusehen:

```
$ lsattr -E -H -l rmt0
attribute      value  description                    user_settable

block_size     1024   BLOCK size (0=variable length)     True
compress       yes    Use data COMPRESSION               True
density_set_1  140    DENSITY setting #1                 True
density_set_2  20     DENSITY setting #2                 True
extfm          yes    Use EXTENDED file marks            True
mode           yes    Use DEVICE BUFFERS during writes   True
```

Dieses 8-mm-Bandlaufwerk nutzt standardmäßig die Datenkomprimierung und eine Blockgröße von 1024 Bytes.

Sie müssen den `chdev`-Befehl verwenden, um die vielen Attribute eines Bandlaufwerks zu ändern. (Diese Einstellungen werden also nicht wie bei den anderen Systemen in den Namen der Gerätedateien codiert.) Der folgende Befehl ändert beispielsweise die Blockgröße auf 1024 Bytes und deaktiviert die Komprimierung und das Retensioning für Laufwerk 1:

```
# chdev -l rmt0 -a block_size=1024 -a compress=no -a ret=no
```

Sichern von Dateien und Dateisystemen

Die meisten Systeme bieten eine Vielzahl unterschiedlicher Utilities an, mit denen Backups durchgeführt werden können. Diese reichen von allgemeinen Archivierungsprogrammen wie `tar` und `cpio` bis hin zu Programmen, die aus mehreren Ebenen bestehende inkrementelle Backup-Schemata auf Basis einzelner Dateisysteme implementieren. Als die größten Bänder nur wenige hundert Megabyte aufnehmen konnten, war die Wahl des richtigen Utilitys für System-Backups noch einfach. `tar` und `cpio` wurden für kleine Sicherungen und schnelle Backups oder Datentransfers verwendet, die »zwischendurch« notwendig waren. Die speziell zu diesem Zweck entwickelten Utilities wurden für System-Backups verwendet, weil sie besondere Fähigkeiten (etwa das automatische Spannen von Bändern und die automatische Durchführung inkrementeller Backups) besaßen, die zur Erledigung des Jobs unbedingt erforderlich waren.

Diese Unterscheidung entfällt zum größten Teil, wenn ein einzelnes Band mehrere Gigabyte an Daten speichern kann. Beispielsweise sind inkrementelle Sicherungen nun weniger wichtig, weil alle wichtigen Daten auf ein oder zwei Bänder passen. Große Bänder haben es auch möglich gemacht, ein System in logisch angeordneten Teilstücken zu sichern, die sich willkürlich über das physikalische Dateisystem verstreuen. Ein erfolgreiches System-Backup kann mit allen Utilities durchgeführt werden, die für Ihr System geeignet zu sein scheinen.

Ein dubioser Ratschlag, der häufig bezüglich der Datensicherung erteilt wird, lautet, die Größe des Dateisystems auf die im System maximal verfügbare Backup-Kapazität zu begrenzen. Multi-Tape-Backups bedeuten hier einfach zu viel Ärger und der Backup-Prozess wird vereinfacht, wenn alle Daten eines Dateisystems auf ein einzelnes Band passen.

Nun ist die Sicherung eines Dateisystems auf ein einzelnes Band sicher bequem, aber ich halte es für einen Fehler, die Planung des Dateisystems in dieser Weise von der aktuellen Kapazität des Sicherungsmediums abhängig zu machen. Die Aufteilung von Platten in viele kleinere Dateisysteme schränkt die Flexibilität bezüglich ihrer Ressourcen ein und dieser Aspekt ist sicher wesentlich wichtiger, als die Komplexität des Backups zu reduzieren. Der Aufbau des Dateisystems muss *alle* Faktoren berücksichtigen, die das System und dessen Effektivität beeinflussen. Wenn bandgroße Backups gewünscht werden, kann man immer noch Skripten schreiben, die das erreichen, wenn die allgemeinen Umstände größere Dateisysteme verlangen.

tar oder cpio reichen häufig aus

In einigen Fällen, insbesondere auf Einzelbenutzersystemen, wird kein ausgefeiltes Backup-Schema benötigt. Vielmehr ist es, da es sich bei Verwalter und Benutzer um die gleiche Person handelt, offensichtlich, welche Dateien wichtig sind und wie oft sie sich ändern. In einem solchen Fall können die einfacheren Befehle `tar` und `cpio` ausreichen, um die wichtigen Dateien periodisch auf Band (oder andere Medien) zu sichern.

Ein typisches Beispiel für diese Situation ist eine Workstation mit Unix, aber die Utilities können auch bei Systemen mit relativ kleinen Mengen an kritischen Daten ausreichen. `tar` und `cpio` haben auch den Vorteil, neben lokalen auch über NFS gemountete Dateisysteme sichern zu können.

Der tar-Befehl

Wir beginnen mit einem einfachen Beispiel. Der folgende `tar`-Befehl sichert alle Dateien unter */home* auf dem Standard-Bandlaufwerk:

```
$ tar -c /home
```

`-c` weist das Programm an, ein Backup-Archiv zu erzeugen.

Die `tar`-Option `-C` (großes C) ist nützlich, um Dateien aus verschiedenen Teilen des Dateisystems in einem einzelnen Archiv unterzubringen. Mit dieser Option wird zuerst das Verzeichnis, das als Argument übergeben wurde, als aktuelles Verzeichnis definiert. Erst dann verarbeitet `tar` alle vorhandenen Pfadnamen. `-C` kann auch mehrmals in einem Befehl auftauchen. Zum Beispiel speichern die folgenden `tar`-Befehle alle Dateien unter den Verzeichnissen */home*, */home2* und */chem/public*:

```
$ tar -cf /dev/rmt1 /home /home2 /chem/public
$ tar -cf /dev/rmt1 -C /home . -C /home2 . -C /chem public
```

Die beiden Befehle unterscheiden sich darin, dass der erste alle Dateien mit den absoluten Pfadnamen (z.B. */home/chavez/.login*) speichert, während der zweite Befehl mit relativen

Pfadnamen (etwa *./chavez/.login*) arbeitet. Die Dateien aus dem ersten Archiv würden also immer an derselben Position im Dateisystem wiederhergestellt werden, während die Daten aus dem zweiten Archiv immer relativ zum aktuellen Verzeichnis wiederhergestellt werden (d.h. mit anderen Worten: immer relativ zu dem Verzeichnis, aus dem heraus der `restore`-Befehl ausgeführt wurde).

Es ist eine gute Idee, absolute Pfadnamen als Argument an `-C` zu übergeben. Relative Pfadnamen werden von `-C` immer mit Bezug auf das Verzeichnis ermittelt, das gerade aktiv war, als die Option bearbeitet wurde. Die Option bezieht sich nicht auf das Verzeichnis, das aktuell war, als der `tar`-Befehl gestartet wurde. Aufeinander folgende `-C`-Optionen werden also akkumuliert, und `tar`-Befehle, die mehrere davon mit relativen Pfadnamen verwenden, werden nahezu uninterpretierbar.

Traditionell wurden alle `tar`-Optionen in einer einzelnen Gruppe direkt hinter dem Befehl angegeben, ohne dass der normale Bindestrich nötig gewesen wäre. Der POSIX-Standard spezifiziert eine etwas traditionellere Unix-Syntax. Hierbei wird die zweite der ersten Form des Befehls vorgezogen:

```
$ tar xpfb /dev/rmt1 1024 ...
$ tar -x -p -f /dev/rmt1 -b 1024 ...
```

Die `tar`-Versionen auf den aktuellen Betriebssystemen akzeptieren in der Regel beide Formate. Möglicherweise wird es in Zukunft aber notwendig sein, zumindest einen Bindestrich zu Beginn anzugeben.

`tar`-Archive sind häufig komprimiert, weshalb Ihnen häufig komprimierte tar-Archive mit Namen wie *datei.tar.Z*, *datei.tar.gz* oder *datei.tgz* begegnen werden (die beiden letztgenannten Dateien wurden mit dem GNU-Utility `gzip` komprimiert).

Solaris-Erweiterungen des tar-Befehls. Die Solaris-Version von `tar` bietet Erweiterungen an, die für Backups auf Systemebene besser geeignet sind. Sie erlauben es, die gesamte oder einen Teil der Liste der zu sichernden Dateien und Verzeichnisse in einer oder mehreren Textdateien unterzubringen (mit jeweils einem Element pro Zeile). Diese Dateien werden in die an `tar` übergebene Dateiliste eingebunden, indem man ihnen ein `-I` voranstellt:

```
$ tar cvfX /dev/rst0 Nicht_Sicher /home -I Andere_Benutzerdateien -I Vermischtes
```

Der Befehl sichert die Dateien und Verzeichnisse aus den beiden Include-Dateien und diejenigen in */home*. Der Befehl macht auch den Einsatz der Option `-X` deutlich, der Sie eine Liste mit Namen und Verzeichnissen übergeben, die von `tar` ignoriert werden. Beachten Sie, dass Wildcards weder in Include- noch in Ausschluss-Dateien erlaubt sind. Im Falle eines Konflikts hat der Ausschluss Vorrang vor der Aufnahme.

Die `-I`- und `-X`-Optionen können auch in Restore-Operationen verwendet werden, die mit `tar` ausgeführt werden.

Auf Solaris und einer Reihe anderer System V-Systeme kann die Datei */etc/default/tar* verwendet werden, um die Abbildung der Standard-Archivziele festzulegen. Diese Ziele werden

bei tar über Codezeichen festgelegt, die aus einer Ziffer bestehen (der Befehl tar 1c legt beispielsweise ein Archiv auf Laufwerk 1 an). Hier ist eine Version eines Solaris-Systems:

```
#                               Block   #
#Archive=Device                 Size    Blocks
#
archive0=/dev/rmt/0             20      0
archive1=/dev/rmt/0n            20      0
archive2=/dev/rmt/1             20      0
archive3=/dev/rmt/1n            20      0
archive4=/dev/rmt/0             126     0
archive5=/dev/rmt/0n            126     0
archive6=/dev/rmt/1             126     0
archive7=/dev/rmt/1n            126     0
```

Der erste Eintrag legt das Gerät fest, das verwendet wird, wenn man tar 0 angibt. In diesem Fall ist es das erste Bandlaufwerk mit seinen Standardmodi. Der zweite Eintrag definiert Archiv 1 als das erste Bandlaufwerk im nicht zurückspulenden Modus. Die beiden restlichen Felder sind optional. Sie legen die Blockgröße des Gerätes und dessen Gesamtkapazität fest (Letzteres kann auf null gesetzt werden, um den Befehl die Ende-des-Mediums-Markierung finden zu lassen).

GNU tar: Linux und FreeBSD. Linux-Distributionen und FreeBSD stellen die GNU-Version des tar-Befehls zur Verfügung. Es unterstützt die üblichen Features zur Anpassung von tar sowie einige Verbesserungen, z.B. die Fähigkeit mehrerer Volumes (-M) und den Einsatz der gzip-Komprimierung (-z). Der folgende Befehl extrahiert beispielsweise den Inhalt des angegebenen komprimierten tar-Archivs:

```
$ tar xfz funsoftware.tgz
```

Der cpio-Befehl

cpio kann ebenfalls für Backups verwendet werden. Es bietet mehrere Vorteile:

- Es wurde entworfen, um willkürlich zusammengewürfelte Dateisätze zu sichern. tar ist für Unterverzeichnisse besser geeignet.
- Es packt die Daten auf dem Band wesentlich effizienter als tar. Wenn es für Sie wichtig ist, alle Daten auf ein Band zu bekommen, dann mag cpio die bessere Wahl sein.
- Beim Wiedereinspielen der Daten überspringt es fehlerhafte Stellen auf dem Band; tar hingegen bricht einfach ab.
- Es kann mit mehreren Bändern arbeiten, während die meisten tar-Versionen mit nur einem Band umgehen können.

Bei Verwendung der Option -o kopiert cpio die Dateien, deren Pfadnamen es über die Standardeingabe (oft über ls oder find) erhält, zur Standardausgabe. Die Standardausgabe lässt sich umlenken, um mittels cpio ein Band oder eine Diskette zu beschreiben. Die folgenden Beispiele zeigen, wie sich cpio für Backups einsetzen lässt:

```
$ find /home -print | cpio -o >/dev/rmt0
```

```
$ find /home -cpio /dev/rmt0
```

Der erste Befehl kopiert alle Dateien im Verzeichnis */home* und seinen Unterverzeichnissen auf das Band in Laufwerk 0. Der zweite Befehl führt ein identisches Backup durch, wobei mit einer find-Version gearbeitet wird, die die Option -cpio unterstützt.

Inkrementelle Backups mit tar und cpio

Die Kombination von find und tar oder cpio stellt eine einfache Möglichkeit dar, inkrementelle Backups durchzuführen. Dies gilt besonders, wenn nur zwei oder drei unterschiedliche Backup-Level benötigt werden. Beispielsweise kopiert der folgende Befehl mit Ausnahme von Objektdateien (*.o*) alle heute modifizierten Dateien unter */home* in ein Archiv auf */dev/rmt1*:

```
$ find /home -mtime -1 ! -name \*.o -print | cpio -o >/dev/rmt1
$ tar c1 `find /home -mtime -1 ! -name `*.o' ! -type d -print`
```

Der find-Befehl, der zusammen mit tar verwendet wird, muss Verzeichnisse ausschließen, weil tar automatisch *jede* Datei unter jedem Verzeichnis archiviert, das in der Dateiliste aufgeführt wird. Alle Verzeichnisse, in denen sich *irgendeine* Datei verändert hat, werden aber von find ausgegeben.

Sie können auch die find-Option -newer verwenden, um ein inkrementelles Backup auf diese Weise durchzuführen:

```
$ touch /backup/home_full
$ find /home -print | cpio -o > /dev/rmt0
Einen Tag später...
$ touch /backup/home_incr_1
$ find /home -newer /backup/home_full -print | cpio -o > /dev/rmt0
```

Der erste Befehl markiert die Datei */backup/home_full* mit dem touch-Befehl (*/backup* ist ein Verzeichnis, das zur zeitlichen Kennzeichnung der Backups erzeugt wurde). Der zweite Befehl führt ein vollständiges Backup auf */home* durch. Irgendwann später können die zweiten Befehle genutzt werden, um alle Dateien zu archivieren, die sich seit dem ersten Backup geändert haben. Indem alle Dateien zeitlich gekennzeichnet werden, bevor die Sicherung beginnt, wird sichergestellt, dass alle Dateien, die während des Schreibens modifiziert werden, beim nächsten Backup gesichert werden. Dabei spielt es keine Rolle, ob die Dateien in diesem Backup berücksichtigt wurden oder nicht.

pax: Frieden zwischen tar und cpio

Der pax-Befehl versucht die Lücke zwischen tar und cpio zu schließen, indem er ein einzelnes allgemein einsetzbares Archivierungs-Utility zur Verfügung stellt.[12] Er kann Archive in beiden Formaten lesen und schreiben (und schreibt standardmäßig tar-Archive) und bie-

12 Tatsächlich sind cpio und tar auf Systemen, die pax verwenden, häufig nur Links auf pax. Die Syntax von pax ist eine Verschmelzung der beiden, was bei einem von POSIX aufgezwungenen Programm nicht weiter verwunderlich ist (auch wenn der Name für *portable archive exchange* steht).

tet beiden gegenüber einige Verbesserungen, was es in vielen Umgebungen zu einem ausgezeichneten Utility für System-Backups macht. pax ist für alle hier betrachteten Unix-Versionen verfügbar. Wie bei cpio können sich auch pax-Archive über mehrere Media-Volumes erstrecken.

Die allgemeine Syntax für pax sieht wie folgt aus:

```
pax [modus_option] andere_optionen zu_sichernde_dateien
```

Die *modus_option* legt fest, ob Dateien in ein Archiv geschrieben oder aus einem Archiv extrahiert werden sollen. Dabei steht -w für das Schreiben und -r für das Lesen und Extrahieren eines Archivs. Mit -rw werden Dateien in ein alternatives Verzeichnis auf der Platte kopiert (entspricht cpio -p). Der Standardmodus von pax, bei dem keine *modus_option* angegeben wird, gibt den Inhalt eines Archivs aus.

Die folgenden Befehle zeigen die pax-Modi im Einsatz:

```
$ pax -w -f /dev/rmt0 /home /chem
$ find /home /chem -mtime -1 -print | pax -w -f /dev/rmt0
$ pax -w -X -f /dev/rmt0 /
```

Die ersten beiden Befehle führen ein vollständiges und ein inkrementelles Backup der Dateien in */home* und */chem* durch. In beiden Fällen wird dabei das Standard-Bandlaufwerk verwendet. Der dritte Befehl sichert alle Dateien in der Plattenpartition, die dem root-Verzeichnis entspricht. Die Option -X weist pax an, Dateisystemgrenzen nicht zu überschreiten.

AIX bevorzugt pax gegenüber dem normalen tar und cpio. Der Befehl wurde so erweitert, dass er große Dateien (über 2 GB) unterstützt.

Sichern individueller Dateisysteme mit dump

Das BSD-Utility dump stellt den nächsten Schritt in der Entwicklung von Backup-Systemen unter Unix dar. Es sichert selektiv alle Dateien innerhalb eines Dateisystems (einer einzelnen Partition), indem es die jeder Inode entsprechenden Daten in das Archiv auf dem Backup-Gerät kopiert. Es hat darüber hinaus den Vorteil, jede Art von Datei sichern zu können, also auch Gerätedateien und Sparse-Files. Obwohl es zwischen den Implementie-

Benutzer zum Anlegen von Backups bewegen

Manche Sites delegieren bestimmte Backup-Verantwortlichkeiten an individuelle Benutzer: Wenn eine Site zu viele Workstations besitzt, um die Sicherung aller lokalen Platten praktikabel erscheinen zu lassen, wenn wichtige Daten auch auf Nicht-Unix-Systemen wie PCs liegen (insbesondere wenn diese nicht an das lokale Netzwerk angeschlossen sind) und so weiter.

Allerdings werden Sie, auch wenn Sie die eigentliche Datensicherung nicht selbst vornehmen, wahrscheinlich dennoch für den technischen Support verantwortlich sein. Und wahrscheinlich werden Sie die Benutzer noch viel häufiger daran erinnern müssen, die Backups auch wirklich durchzuführen. Hier einige Ansätze, mit denen ich versucht habe, das zu erreichen:

- Gewöhnen Sie es sich an, die Benutzer zu ermuntern, statt ihnen zu drohen (Drohungen funktionieren sowieso nicht).
- Nutzen Sie sanften Druck zu Ihrem Vorteil. Die Einrichtung einer zentralen Ablage für die Backups, die Sie sich regelmäßig ansehen, macht deutlich, wer die ihm zugedachten Backups anlegt und wer nicht. Beachten Sie, dass diese Idee ungeeignet ist, wenn sensitive Daten im Spiel sind.
- Entwickeln Sie Werkzeuge, die den Backup-Prozess für die Benutzer so weit wie möglich automatisieren. Jeder hat die Zeit, vor dem Nach-Hause-Gehen ein Band einzulegen und ein Skript zu starten.
- Stellen Sie eine zentrale Sammelstelle für wichtige Schlüsseldateien zur Verfügung, die als Teil der System/Site-Prozedur gesichert werden. Benutzer können Schlüsseldateien kopieren und wissen, dass diese wirklich gesichert werden, wenn sie in der Patsche sitzen und wirklich keine Zeit haben, die Backups selbst anzulegen.

rungen für die verschiedenen Unix-Varianten kleinere Unterschiede gibt, gilt das im Folgenden Gesagte für:

AIX	`backup`
FreeBSD	`dump`
HP-UX	`dump` und `vxdump`
Linux	`dump` (aber das Paket ist standardmäßig normalerweise nicht installiert)
Solaris	`ufsdump`
Tru64	`dump` und `vdump`

Auf Systemen, die mehrere Dateisystemtypen unterstützen, könnte `dump` auf UFS-Dateisysteme (BSD) beschränkt sein. Bei Linux-Systemen ist es momentan auf ext2/ext3-Dateisysteme beschränkt, auch wenn das XFS-Dateisystem das vergleichbare `xfsdump`-Utility zur Verfügung stellt. Unter HP-UX unterstützen `vxdump` und `vxrestore` die VxFS-Dateisysteme. Tru64 stellt `vdump` für AdvFS-Dateisysteme zur Verfügung.

dump hält nach, wann es jedes Dateisystem zuletzt gesichert hat und auf welchem Level diese Sicherung erfolgt ist. Diese Information wird in der Datei */etc/dumpdates* festgehalten (außer auf HP-UX-Systemen, die mit */var/adm/dumpdates* arbeiten). Ein typischer Eintrag in dieser Datei sieht wie folgt aus:

```
/dev/disk2e       2 Sun Feb  5 13:14:56 1995
```

Dieser Eintrag besagt, dass das Dateisystem */dev/disk2e* zuletzt am Sonntag, dem 5. Februar, während eines Level-2-Backups gesichert wurde. Findet dump kein Dateisystem in dieser Liste, geht es davon aus, dass es noch nie gesichert wurde.

Wenn die Datei *dumpdates* nicht existiert, legt der folgende Befehl sie an:

```
# touch /path/dumpdates
```

Die *dumpdates*-Datei muss dem Benutzer *root* gehören. Wenn die Datei nicht existiert, legt dump sie nicht an, und Dateisystem-Backups werden nicht festgehalten. Sie sollten diese Datei also anlegen, bevor Sie dump das erste Mal gespeichert.

Der dump-Befehl hat zwei allgemeine Formen:

```
$ dump Optionen-mit-Argumenten Dateisystem
$ dump Optionsbuchstaben Argumente Dateisystem
```

Dateisystem ist die blockorientierte Gerätedatei, die dem zu sichernden Dateisystem bzw. dem Mountpunkt aus der Dateisystem-Konfigurationsdatei entspricht. Bei der ersten (neueren) Form ist das erste Element eine Liste der beim Backup zu verwendenden Optionen. Die Argumente folgen in der üblichen Art unmittelbar auf den Optionsbuchstaben (z.B. `-f /dev/tape`).

Bei der zweiten (älteren) Form ist *Optionsbuchstaben* eine Liste mit Buchstaben, die den gewünschten Optionen entsprechen, und *Argumente* sind die zu jeder Option gehörenden Argumente (in der gleichen Reihenfolge). Diese Syntax ist unter Solaris und HP-UX immer noch die einzig verfügbare.

Nicht alle Optionen verlangen nach Argumenten. Trotzdem müssen die Argumente *exakt* in der Reihenfolge und Anzahl mit den Optionen, die nach Argumenten verlangen, übereinstimmen. Betrachten Sie beispielsweise die Optionen `0sd`. Die Optionen `s` und `d` verlangen nach Argumenten, `0` nicht. Ein dump-Befehl, der diese Optionen verwendet, muss daher die folgende Form haben:

```
$ dump 0sd s-Argument d-Argument Dateisystem
```

Hält man sich nicht an diese Regel, kann das schmerzhafte Konsequenzen haben, wenn man den Befehl als *root* ausführt. Sie können dann beispielsweise das Dateisystem zerstören, wenn Sie das Argument der `f`-Option und das letzte Argument von dump vertauschen. Sie werden von mir kein Gegenargument hören, wenn Sie jetzt anmerken, dass es sich hier um einen Entwurfsfehler handelt, der schon längst hätte behoben werden müssen. Stellen Sie sicher, dass Sie für jede Option, die nach einem Argument verlangt, ein solches angeben, wenn Sie mit dump arbeiten. Um solche Bedienungsfehler zu vermeiden, sollten Sie Shell-Skripten anlegen, die dump automatisch mit den richtigen Optionen aufrufen.

Die wichtigsten dump-Optionen sind (wir verwenden die neuere Form):

`-0, ..., -9`
: Die Ziffern 0 bis 9 bezeichnen den Backup-Level, der von diesem Befehl angelegt wird. Für einen gegebenen Level *n* durchsucht `dump` die Datei */etc/dumpdates* nach einem Eintrag, der das Datum des letzten Backups des Dateisystems mit dem Level *n-1* oder niedriger enthält. `dump` sichert dann alle Dateien, die sich seit dem letzten Backup geändert haben. Ist *n* gleich null, sichert `dump` das gesamte Dateisystem. Enthält */etc/dumpdates* keinen Backup-Eintrag mit Level *n-1* oder niedriger, sichert `dump` ebenfalls das gesamte Dateisystem. Wird kein Level angegeben, führt `dump` ein Level-9-Backup aus. Diese Option erfordert kein Argument.

  Ältere `dump`-Versionen, die mit Bindestrichen arbeitende Optionen nicht unterstützen, verlangen das Level als erste Option.

`-u`
: Wenn `dump` erfolgreich war, bringt diese Option die Datei */etc/dumpdates* auf den neuesten Stand. Sie benötigt kein Argument.

`-f` *`Gerät`*
: Diese Option besagt, dass das Backup nicht auf das voreingestellte Bandlaufwerk (also z. B. in eine Datei oder auf ein anderes Gerät) erfolgen soll. Die Standardgeräte der verschiedenen Unix-Versionen wurden ja bereits erwähnt. Wenn Sie diese Option verwenden, müssen Sie ein Argument angeben, und das Argument muss unbedingt vor der Gerätedatei stehen. Das Argument »-« (der Bindestrich) steht für die Standardausgabe.

`-W`
: Gibt aus, was beim Aufruf des Befehls gesichert werden würde, führt die eigentliche Sicherung aber nicht durch.

`-s` *`Fuß`* `-d` *`Dichte`*
: Diese Optionen wurden bei älteren `dump`-Versionen benötigt, um die Kapazität des Backup-Mediums zu bestimmen. Neue `dump`-Versionen benötigen sie grundsätzlich nicht, weil sie so lange weiterschreiben, bis sie die Medienende-Markierung erkennen.

  Falls Sie `dump` über die Bandlänge etwas vorgaukeln müssen, weil Ihre Version standardmäßig ein Kapazitätslimit verwendet, das sich an veralteten 9-Spur-Laufwerken orientiert, können Sie mit `-s` die *Größe* des Backup-Bandes in englischen Fuß angeben. `-d` legt die Dichte des Bandes in Bits pro Zoll fest. Weil `dump` die Mediumende-Markierung erkennt, bevor es das eigentliche Limit erreicht, können Sie solche Probleme einfach dadurch lösen, dass Sie die Kapazität auf einen Wert setzen, der weit über dem eigentlichen Limit liegt. So definieren die Optionen `-d 50000 -s 90000` eine Bandkapazität von etwas über 4 GB.

`-b` *`faktor`*
: Legt die für das Band zu verwendende Blockgröße in Einheiten von 1024-Byte-Blöcken (manchmal 512-Byte-Blöcken) fest.

Hier eine typische Anwendung des `dump`-Befehls:

```
$ dump -1 -u -f /dev/tape /chem
```

Dieser Befehl führt ein inkrementelles Level-1-Backup des */chem*-Dateisystems auf das mit */dev/tape* verknüpfte Bandlaufwerk durch. `dump` aktualisiert die *dumpdates*-Datei nach getaner Arbeit.

`dump` benachrichtigt den Benutzer, wenn eine Interaktion notwendig ist. Meistens hat `dump` einfach das Band voll geschrieben und verlangt nach einem neuen. Außerdem fragt `dump` bei Problemen nach, ob es versuchen soll, diese zu beseitigen. Zusätzlich liefert `dump` Rückmeldungen darüber, was es gerade tut und wie viele Bänder und wie viel Zeit für das Backup voraussichtlich benötigt werden.

Das HP-UX-Utility fbackup

HP-UX stellt für System-Backups die Utilities `fbackup` und `frecover` zur Verfügung. Ein signifikanter Vorteil, den sie gegenüber den Standard-Unix-Utilities besitzen, besteht darin, dass sie HP-UX-Zugriffskontroll-Listen zusammen mit anderen Datei-Metadaten sichern und wiederherstellen können.

`fbackup` ermöglicht, genau wie `dump`, inkrementelle Backups in 9 Levels. `fbackup` speichert Backup-Records in der Datei */var/adm/fbackupfiles/dates*. Diese Datei muss vom Systemadministrator angelegt werden, bevor `fbackup` verwendet wird.

Die folgenden Beispiele machen deutlich, wie `fbackup` für System-Backups eingesetzt werden könnte:

```
# fbackup -0u -f /dev/rmt/1m -i /chem
# fbackup -1u -i /chem -i /bio -e /bio/med
# fbackup -1u -f /dev/rmt/0m -f /dev/rmt/1m -i /chem
# fbackup -0u -g /backup/chemists.graph -I /backup/chemists.TOC
```

Der erste Befehl führt ein vollständiges Backup von */chem* auf Bandlaufwerk 1 durch und aktualisiert die `fbackup`-Datenbank. Der zweite Befehl führt ein Level-1-Backup von */chem* und */bio* durch, schließt dabei aber das Verzeichnis */bio/med* aus (Sie können so viele `-i`- und `-e`-Optionen wie nötig aufnehmen). Der dritte Befehl führt ein Level-1-Backup von */chem* durch und verwendet dabei mehrere Laufwerke hintereinander.

Der letzte Befehl führt ein vollständiges Backup entsprechend den Vorgaben der so genannten Graph-Datei */backup/chemists.graph* durch. Ein Index des Backups wird dabei in die Datei */backup/chemists.TOC* geschrieben. Eine Graph-Datei ist eine Textdatei mit dem folgenden Format:

```
c    pfad
```

c ist dabei ein Code, der angibt, ob der *pfad* im Backup aufgenommen (`i` für include) oder vom Backup ausgeschlossen (`e` für exclude) werden soll.

Verwandte Utilities

Es gibt noch zwei weitere Unix-Utilities, die Sie kennen sollten, weil sie gelegentlich ebenfalls für Backups verwendet werden.

Daten mit dd kopieren und konvertieren

Das `dd`-Utility transferiert Rohdaten zwischen Geräten. Es ist für die Konvertierung von Daten zwischen einzelnen Systemen und für das Lesen und Schreiben von Daten zu/von Rechnern ohne Unix geeignet. Als Argumente werden eine Reihe von *`Option=Wert`*-Paaren übergeben. Einige der nützlichsten Optionen sind:

`if` Input file: Quelle der Daten.

`of` Output file: Ziel der Daten.

`ibs`
: Blockgröße der Eingabe in Bytes (voreingestellt sind 512).

`obs`
: Blockgröße der Ausgabe in Bytes (voreingestellt sind 512).

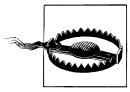
`fskip`
: Daten überspringen, bevor Daten übertragen werden (nicht bei allen Implementierungen vorhanden).

`count`
: Anzahl der zu übertragenden Blöcke.

`conv`
: Schlüsselwort/-wörter, die die Konvertierung der Eingabe vor der Ausgabe spezifizieren: `swab` bedeutet, dass die Bytes getauscht (»geswappt«) werden sollen, und ist der am häufigsten verwendete Konvertierungstyp. `lcase` und `ucase` sorgen für die Umwandlung in Klein-/Großbuchstaben. `ascii` und `ebcdic` konvertieren die Daten in ASCII bzw. EBCDIC.

Zum Beispiel verarbeitet der folgende Befehl die dritte Datei des Bandes in Laufwerk 0 mit einer Eingabe-Blockgröße von 1024 Bytes, vertauscht dabei alle Datenbytes und schreibt die konvertierte Ausgabe in die Datei */chem/data/c70o.dat*:

```
$ dd if=/dev/rmt0 of=/chem/data/c70o.dat \
     ibs=1024 fskip=2 conv=swab
```



Wie immer müssen Sie bei der Angabe der richtigen Geräte für `if` und `of` sehr vorsichtig sein. Sie zu vertauschen kann furchtbare Konsequenzen haben.

Datenbänder auf Geräteebene mit mt bearbeiten

Unix bietet den Befehl `mt` für die direkte Bearbeitung von Datenbändern. Sie können damit das Band ausrichten (um beispielsweise Backups zu überspringen), Bänder zurückspulen und ähnliche grundlegende Bandoperationen ausführen. Die Syntax lautet:

```
$ mt [-f Bandeinheit] Befehl
```

Bandeinheit steht für das zu verwendende Bandlaufwerk und *Befehl* für die gewünschte Aktion. Nützliche Schlüsselwörter sind `rewind` (zum Rückspulen des Bandes), `status` (für die Ausgabe des Gerätestatus – Sie können etwa sehen, ob das Gerät benutzt wird), `fsf` *n* (zum Überspringen der nächsten *n* Dateien) und `bsf` *n* (*n* Dateien zurück).

Um beispielsweise das Band im zweiten Laufwerk zurückzuspulen, müssen Sie den folgenden Befehl benutzen:

```
$ mt -f /dev/rmt1 rewind
```

Die Solaris-Version von `mt` kennt auch einen `asf`-Befehl, der das Band zur *n*-ten Datei auf dem Band bewegt (*n* wird dabei `asf` als Argument übergeben), wobei die aktuelle Position des Bandes keine Rolle spielt.

Unter FreeBSD wird der `mt`-Befehl verwendet, um Dichte und Komprimierung des Laufwerks einzustellen:

```
$ mt -f /dev/nrsa0 comp on density 0x26
```

AIX enthält das `tctl`-Utility (`mt` ist in Wirklichkeit nur ein Link darauf). `tctl` verwendet die gleiche Syntax wie `mt` und bietet darüber hinaus einige selten benötigte Operationen.

Dateien aus Backups wiederherstellen

Alle in den vorangegangenen Abschnitten beschriebenen Backup-Einrichtungen besitzen entsprechende Einrichtungen zur Wiederherstellung der Dateien. Wie werden uns in diesem Abschnitt jede dieser Einrichtungen ansehen.

Dateien aus tar- und cpio-Archiven wiederherstellen

Einzelne Dateien oder gesamte Unterbäume können aus `tar`- und `cpio`-Archiven auf sehr einfache Weise wiederhergestellt werden. Zum Beispiel stellen die folgenden Befehle die Datei */home/chavez/freeway/quake95.data* und das Home-Verzeichnis des Benutzers *harvey* aus einem Archiv wieder her, das vom Verzeichnis */home* auf einem Band in Laufwerk */dev/rmt0* angelegt wurde:

```
$ tar -xp /home/chavez/freeway/quake95.data
$ cpio -im '*quake95.data' < /dev/rmt0
$ tar -xp /home/harvey
$ cpio -imd '/home/harvey*' < /dev/rmt0
```

Die `tar`-Option `-p` und die `cpio`-Option `-m` stellen sicher, dass alle Dateien mit den jeweils korrekten Dateiattributen wiederhergestellt werden. Die `cpio`-Option `-d` sorgt bei Bedarf für die Generierung von Unterverzeichnissen, wenn ganze Verzeichnisbäume wieder hergestellt werden (`tar` macht das standardmäßig).[13]

Eine Wiederherstellung mit `pax` verläuft ähnlich. Der erste der nachfolgenden Befehle gibt zum Beispiel eine Liste mit den Dateien aus, die auf dem Band in Laufwerk 0 enthalten sind, während die restlichen Befehle verschiedene Dateien daraus extrahieren:

```
$ pax -f /dev/rmt0 -v          –v  liefert ein etwas detaillierteres Listing.
$ pax -r '/h95/*.exe'          Dateien mit Hilfe eines regulären Ausdrucks auswählen.
```

13 Der zweite `cpio`-Befehl geht außerdem davon aus, dass außer dem Home-Verzeichnis des Benutzers *harvey* keine weiteren Dateien mit »harvey« beginnen.

```
$ pax -r /home/chavez                  Home-Verzeichnis von chavez wiederherstellen.
$ pax -r -f my_archive -c '*.o'        Alles außer Objektdateien wiederherstellen.
# pax -r -pe -f /dev/rmt0              Dateien mit Eigentümer, Modus und Mod.-Zeit
                                       wiederherstellen.
```

Das coolste pax-Feature ist sicherlich seine -s-Option, mit der Sie Dateinamen bearbeiten können, während die Dateien geschrieben, extrahiert oder auch nur aus dem Archiv aufgelistet werden. Die Option verlangt einen Substitutionsbefehl wie bei ed oder sed als Argument (der üblicherweise in einfachen Anführungszeichen eingeschlossen sein muss). Dieser Substitutionsbefehl gibt an, wie die Dateinamen transformiert werden sollen. Der folgende Befehl ändert zum Beispiel die Verzeichnisnamen für jede Datei auf der zweiten Ebene von *chavez* in *harvey*, während die Dateien gelesen werden, und ändert damit das Ziel auf der Festplatte:

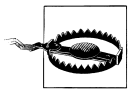
```
$ pax -r -s ',^/home/chavez/,/home/harvey/,' \
      -f /dev/rmt0 /home/chavez
```

Die Substitutionsklausel sucht am Anfang jedes wiederherzustellenden Pfadnamens nach */home/chavez* und ändert ihn in */home/harvey*. Der Substitutionsstring verwendet Kommas als Trennzeichen für die einzelnen Felder.

Hier einige weitere -s-Klauseln für spezielle Transformationen:

```
-s ',^/home/chavez/,,'     Verzeichniskomponenten teilweise entfernen.
-s ',^.*//*,,'             Gesamte Verzeichniskomponente entfernen.
-s ',^//*,,'               Pfadnamen relativ zum aktuellen Verzeichnis machen.
```

Mehrere -s-Optionen sind erlaubt, aber nur die erste zutreffende wird für einen gegebenen Dateinamen verwendet.



Beachten Sie, dass auch pax ein wenig exzentrisch sein kann. Eine der ärgerlichsten Eigenheiten ist die Tatsache, dass bei einigen pax-Versionen über Wildcards erkannte Verzeichnisse bei Restore-Operationen nicht vollständig wiederhergestellt werden, sondern nur explizit aufgeführte Verzeichnisse. Beachten Sie, dass das genau im Gegensatz zur Funktionsweise von cpio steht und auch der Arbeitsweise von tar widerspricht. Ich würde es als Bug sehen, wenn er nicht in mehr als einer Herstellerversion (wenn auch nicht in jeder) auftauchen würde. Bei der Verwendung von pax ist Vorsicht angebracht.

Wiederherstellung aus dump-Archiven

Das Utility restore spielt Dateien wieder ein, die auf Backup-Bändern liegen, die mit dump angelegt wurden. Es wird von allen Systemen unterstützt, die über eine Version von dump verfügen. Auf Solaris heißt es allerdings ufsrestore, um eine einheitliche Namensgebung für dessen dump-Version beizubehalten. HP-UX und Tru64 stellen die Befehle vxrestore und vrestore für ihre Standard-Dateisystemtypen zur Verfügung. All diese Befehle haben die gleiche Syntax und die gleichen Optionen. Mit diesen Befehlen können einzelne Dateien, Verzeichnisse oder vollständige Dateisysteme wiederhergestellt werden.

Um ein gesamtes Dateisystem wiederherzustellen, müssen Sie die aktuellsten Backup-Bänder *jedes* Levels wieder einspielen, d.h. die des letzten vollständigen Dumps (0), des letzten Level-1-Dumps usw. Sie müssen jeden Level (beginnend bei 0) nacheinander wiederherstellen. `restore` speichert die gelesenen Dateien im aktuellen Arbeitsverzeichnis. Um also ein ganzes Dateisystem wiederherzustellen, sollten Sie ein neues, leeres Dateisystem mounten, in das Verzeichnis wechseln, an dem das Dateisystem gemountet wurde, und erst dann die Dateien mit `restore` wiederherstellen. Beachten Sie, dass solche `restore`-Operationen den Nebeneffekt haben, gelöschte Dateien wiederherzustellen.

Nach einem vollständigen Restore müssen Sie ein vollständiges Backup (Level 0) durchführen. Der Grund dafür ist der, dass `dump` die Dateien mit ihren internen Inode-Nummern zurückspielt. Die Dateien, die Sie gerade zurückgespielt haben, werden mit den Inodes des Dateisystems nicht mehr übereinstimmen (die während des Wiedereinspielens sequenziell vergeben werden).

Ganz allgemein hat der `restore`-Befehl die folgenden Formen (wie bei `dump`):

```
$ restore Optionen-mit-Argumenten [Dateien-und-Verzeichnisse]
$ restore Optionen Argumente [Dateien-und-Verzeichnisse]
```

Dateien-und-Verzeichnisse ist eine Liste der Dateien und Verzeichnisse, die `restore` vom Sicherungsband wiederherstellen soll. Werden keine Dateien angegeben, wird das gesamte Band wiederhergestellt.

Bei der ersten (neueren) Form ist das erste Element die Liste der Optionen, die für dieses Backup verwendet werden sollen. Die Argumente folgen wie üblich unmittelbar auf die Optionsbuchstaben (z.B. `-f /dev/tape`). Bei der zweiten (älteren) Form ist *Optionen* eine Liste mit den Buchstaben der gewünschten Optionen und *Argumente* die Liste der mit jeder Option verknüpften Werte (in der gleichen Reihenfolge). Diese Syntax ist unter AIX und Solaris nach wie vor die einzig verfügbare.

Die meisten `restore`-Optionen haben keine Argumente. Genau wie bei `dump` ist es aber wichtig, dass die benötigten Argumente in der Reihenfolge erscheinen, die von den Optionen erwartet wird.

`restore` schreibt die gelesenen Dateien in das aktuelle Arbeitsverzeichnis. Wird ein Verzeichnis für die Wiederherstellung gewählt, stellt `restore` sowohl das Verzeichnis als auch alle darin enthaltenen Dateien wieder her, gleichgültig ob die (nachfolgend noch beschriebene) Option `-h` angegeben wurde oder nicht.

Die wichtigsten `restore`-Optionen sind:

`-r`

: *Lies (Read)* das gesamte Band und spiele es wieder ein. Dies ist ein sehr mächtiger Befehl. Er sollte nur verwendet werden, um vollständige Dateisysteme wiederherzustellen, die sich auf einem oder mehreren Bändern befinden. Das Dateisystem, in das das Band eingelesen wird, sollte neu angelegt werden und völlig leer sein. Die Option kann auch verwendet werden, um einen inkrementellen Dump auf ein gerade neu wiederhergestelltes Dateisystem aufzuspielen. Nachdem Sie mit `-r` also einen voll-

ständigen Dump wiederhergestellt haben, verwenden Sie den Befehl erneut, um nacheinander die inkrementellen Dumps des Dateisystems einzuspielen, bis das Dateisystem vollständig wiederhergestellt ist.

`-x`
: *Extrahieren* aller angegebenen Dateien und Verzeichnisse und Wiederherstellung im aktuellen Verzeichnis. Jeder zu extrahierende Dateiname muss als vollständiger Pfadname *relativ* zum root-Verzeichnis des wiederherzustellenden Dateisystems angegeben werden. Um beispielsweise die Datei */chem/pub/old/gold.dat* aus einem Dump des */chem*-Dateisystems wiederherzustellen, müssen Sie den Dateinamen als *pub/old/gold.dat* angeben. Sie sollten sich in */chem* befinden, wenn Sie den `restore`-Befehl ausführen und die Datei an ihrer ursprünglichen Position wieder eingespielt werden soll.

`-t`
: Gibt die Namen der aufgeführten Dateien und Verzeichnisse auf dem Terminal aus, wenn sie auf dem Backup-Band vorhanden sind (Type). Mit dieser Option können Sie schneller herausfinden, ob eine bestimmte Datei auf einem Band vorhanden ist. Wenn Sie diese Option ohne Dateiliste verwenden, prüft sie, ob das `dump`-Band gelesen werden kann.

`-f` *`Datei`*
: Das dazugehörige Argument ist der Name der Datei oder des Geräts, die bzw. das den Dump enthält. Wird das Argument weggelassen, geht `restore` davon aus, dass sich das Band im Standard-Bandlaufwerk befindet. Wird ein Gerät angegeben, liest `restore` den Dump vom entsprechenden Gerät. Sie können für *Datei* den Bindestrich angeben, um die Standardeingabe festzulegen.

`-s` *`n`*
: Der Wert *n* bestimmt, welche Datei des Bandes für die Wiederherstellung verwendet werden soll. Zum Beispiel bedeutet `-s 3`, dass die dritte Datei auf dem Band verwendet werden soll.

`-i`
: Programm im *interaktiven* Modus ausführen. Dies ist immer die bequemste Möglichkeit, kleine Gruppen von Dateien wiederherzustellen. Dieser Modus wird im nächsten Teilabschnitt beschrieben.

Eine typische Anwendung des `restore`-Befehls sieht wie folgt aus:

```
# cd /home
# restore -x -f /dev/rmt1 chavez/mystuff others/myprogram
```

Dieser Befehl spielt das Verzeichnis */home/chavez/mystuff* und die Datei */home/others/myprogram* wieder von Band ein (wobei wir davon ausgehen, dass */home* das Dateisystem im Archiv ist). Es wird im aktuellen Verzeichnis nach den Verzeichnissen *chavez* und *others* gesucht (falls nötig, werden sie angelegt), und dann werden das angegebene Verzeichnis und die entsprechende Datei wiederhergestellt. Beide standen ursprünglich im Verzeichnis */home*. Beachten Sie, dass der Name des Mountpunkts in diesem `restore`-Befehl nicht ver-

wendet wird. Der Befehl muss aus */home* heraus aufgerufen werden, um die Dateien an ihren ursprünglichen Positionen wiederherzustellen.

Auf Solaris- und HP-UX-Systemen lauten die entsprechenden Optionen:

```
xf /dev/rmt1 chavez/mystuff others/myprogram
```

`dump` und `restore` schreiben die Dateien unabhängig davon, wo das Dateisystem gerade aufgesetzt ist, d.h., die Pfadnamen werden relativ zur Position des *eigenen* Dateisystems, nicht des System-Dateisystems verwendet. Das macht auch Sinn, weil das Dateisystem überall im Verzeichnisbaum aufgesetzt sein könnte, und die wiederhergestellten Dateien könnten dennoch an ihre korrekte Stelle relativ zum Mountpunkt geschrieben werden.

Wenn Sie Dateien wiederherstellen müssen, die versehentlich zerstört wurden, wird Ihr größtes Problem darin bestehen herauszufinden, welche Bänder diese Dateien enthalten. Sie werden viel Zeit damit verbringen, darauf zu warten, dass das System ein oder mehrere Bänder durchsucht. Wenn Sie mit inkrementellen Backups arbeiten, hilft es Ihnen beträchtlich weiter, wenn Sie wissen, wann eine Datei das letzte Mal modifiziert wurde. Das Anlegen von Online-Inhaltsverzeichnissen ist ebenfalls sehr nützlich (wir besprechen dieses Thema später in diesem Kapitel).

Der interaktive Modus von restore

Der interaktive Modus wird mit der `restore`-Option `-i` gestartet. Sobald Sie sich in diesem Modus befinden, können Sie sich den Inhalt eines Bandes ansehen und Dateien für die Wiederherstellung auswählen. Die Verwendung dieses Modus wird im folgenden Beispiel deutlich:

```
$ restore -i -f /dev/rmt1          Starten im interaktiven Modus.
restore > help
Available commands are:
   ls [arg] - list directory
   cd arg - change directory
   add [arg] - add `arg' to list of files to be extracted
   delete [arg] - delete `arg' from list of files to be extracted
   extract - extract requested files
...
If no `arg' is supplied, the current directory is used
restore > ls                       Verzeichnis auf Band ausgeben.
chavez/    harvey/     /ng
restore > cd chavez/vp             Aktuelles Verzeichnis wechseln.
restore > ls
v_a.c      v_a1.c      v_b3.c      v_d23.c      v_early
restore > add v_a1.c               Dateien auswählen (markieren), die wiederhergestellt werden sollen.
restore > add v_early
restore > ls
v_a.c     *v_a1.c      v_b3.c      v_d23.c     *v_early
restore > delete v_early           Markierung einer Datei wieder aufheben.
restore > extract                  Markierte Dateien in aktuelles Verzeichnis auf der Festplatte schreiben.
You have not read any tapes yet.
Unless you know which volume your file(s) are on you should start
with the last volume and work towards the first.
```

```
Specify next volume #: 1          Bandnummer (wenn bekannt).
set owner/mode for '.'? [yn] n    Eigentumsrechte und Schutz von ./'s nicht ändern.
restore > quit                    Interaktiven Modus beenden.
```

Mit dem letzten Prompt fragt `restore`, ob die Eigentums- und Schutzrechte des aktuellen Verzeichnisses an die des root-Verzeichnisses auf dem Band angepasst werden sollen. Beantworten Sie diese Frage nur mit Ja, wenn Sie ein ganzes Dateisystem wiederherstellen.

Mehrere Backups auf einem Band zusammenfassen

Wollen Sie mehrere Archive auf dem gleichen Band unterbringen, müssen Sie das Band vor dem Schreiben des ersten Archivs zurückspulen (wenn nötig) und dann ein nicht zurückspulendes Gerät für alle nachfolgenden Backup-Operationen verwenden.

Um Dateien von einem mehrere Archive enthaltenden Band wieder einzuspielen, müssen Sie das Band zuerst an der richtigen Stelle positionieren, bevor Sie den Befehl zur Wiederherstellung geben. `restore` kann das mit der Option `-s` automatisch tun. Sie übergeben dabei die Nummer des Archivs als Argument.

Bei allen anderen Backup-Typen positionieren Sie das Band mit dem `mt`-Befehl. Die folgenden Befehle positionieren das Band beispielsweise unmittelbar hinter dem zweiten Archiv auf dem Band:

```
$ mt -f /dev/rmt0 rewind      Falls nötig
$ mt -f /dev/nrmt0 fsf 2
```

Auch hier müssen Sie wieder die nicht zurückspulende Variante des Bandlaufwerks verwenden, da das Band andernfalls nach der Positionierung wieder zurückgespult wird. Einmal am gewünschten Punkt angelangt, können Sie ganz nach Bedarf ein zusätzliches Backup auf Band schreiben oder eine restore-Operation mit dem nächsten Archiv auf dem Band vornehmen.

Das HP-UX-Utility frecover

Das HP-UX-Utility `frecover` spielt Dateien wieder ein, die mit dem `fbackup`-Utility archiviert wurden. Die Syntax ist dabei sehr ähnlich. Der erste der beiden folgenden Befehle stellt zum Beispiel den Unterverzeichnis-Baum */chem/fullerenes* wieder her:

```
# frecover -x -i /chem/fullerenes
# frecover -r -f /dev/rmt/1m
```

Der zweite Befehl spielt alle Dateien des Bandes in Laufwerk 1 wieder ein. `frecover` erkennt auch die Optionen `-i`, `-e` und `-g`. Weitere nützliche Optionen sind:

- `-X` und `-F` spielen alle Dateien relativ zum aktuellen Verzeichnis (absolute Pfade werden in relative umgewandelt) bzw. in das aktuelle Verzeichnis (alle Pfade werden entfernt) wieder ein.
- `-o` legt fest, dass Dateien auf der Platte überschrieben werden sollen, die neueren Datums sind als im Backup.

- `-N` legt fest, dass das Backup-Medium gelesen werden soll, ohne irgendwelche Dateien wiederherzustellen. Auf diese Weise können Sie die Integrität des Backups prüfen und Dateien mit Inhaltsverzeichnissen anlegen.

Daten zwischen Systemen bewegen

Generell legen `tar`, `cpio` und `dump` Archive an, die von vielen verschiedenen Computern gelesen werden können. Manchmal kann es aber Schwierigkeiten geben, wenn Sie ein Band lesen wollen, das nicht von Ihrem System stammt. Für solche Probleme gibt es in der Regel vier Hauptgründe:

Unterschiedliche Blockgrößen
: Den einfachsten Problemfall beim Lesen von Bändern stellen verschiedene Blockgrößen dar, d.h., das Band wurde mit einer anderen Blockgröße geschrieben, als dies vom lesenden Laufwerk erwartet wird. Einige Bandlaufwerke gehen von festen Blockgrößen aus. Sie können die Blockgröße bei Backup- und Restore-Utilities häufig angeben (gängig ist hier die Option `-b`), und bei vielen Systemen können Sie auch die Eigenschaften des Laufwerks selbst einstellen. Die am häufigsten verwendeten Blockgrößen sind 512 und 1024.

Inkompatibilitäten der Archiv-Formate
: Die von frühen Unix-Versionen bereitgestellten Backup-Utilities unterscheiden sich von denen, die heutzutage verwendet werden. Sehr alte Computer sind möglicherweise nicht in der Lage, Bänder zu lesen, die mit aktuellen Maschinen geschrieben wurden. Die modernen Versionen der meisten Utilities bieten Optionen an, die die Rückwärtskompatibilität sicherstellen. Mit diesen Optionen können Sie Bänder auf älteren Formaten lesen.

Unterschiedliche Byteordnungen
: Ob ein Computer mit *Big Endian* oder *Little Endian* arbeitet, bestimmt, wie einzelne Bytes in größeren Dateneinheiten wie Wörtern interpretiert werden. Big Endian-Systeme betrachten das Byte an der niedrigsten Adresse als das signifikanteste, während Little Endian-Systeme es als das am wenigsten signifikante betrachten. Archive auf Datenbändern spiegeln dieses grundlegende Hardwareattribut, genau wie alle anderen Daten des Computers, wider. Wenn Sie Bänder lesen wollen, die auf Computern des einen Typs geschrieben wurden, während Ihr System mit dem anderen Typ arbeitet, dann müssen Sie die Byteordnung vertauschen, bevor Utilities wie `tar` etwas Sinnvolles mit dem Archiv anstellen können.

  Zum Beispiel könnten Sie den folgenden AIX-Befehl verwenden, um sich den Inhalt eines Bandes anzusehen, das auf einem IRIX-System geschrieben wurde:

  ```
  $ dd if=/dev/rmt1 conv=swab | tar tvf -
  ```

  Der `dd`-Befehl liest das Band, vertauscht dabei die Byteordnung und übergibt das so konvertierte Archiv an `tar`, das den Inhalt des gefundenen Archivs auf der Standardausgabe ausgibt. Sie könnten die umgekehrte Reihenfolge verwenden, um ein Archiv mit einer entsprechenden Byteordnung auf Band anzulegen.

Komprimierte Archive

Wenn Sie auf ein Laufwerk schreiben, auf dem automatisch komprimiert wird, werden Sie nicht in der Lage sein, dieses Band in einem Laufwerk zu lesen, das diese Fähigkeit nicht besitzt. Damit Laufwerke ohne Komprimierung die Daten lesen können, müssen Sie das Backup-Utility mit der Gerätedatei verwenden, die für nicht komprimierende Speicherung steht (Details finden Sie in der früheren Erläuterung zu Gerätedateien und in den entsprechenden Manpages Ihres Systems).

Dateien mit Inhaltsverzeichnissen erstellen

Es ist häufig sehr angenehm, Online-Listings des Inhalts der System-Sicherungsbänder zu besitzen. Auf diese Weise wird es einfacher herauszufinden, welche Bänder die von Ihnen benötigten Dateien enthalten (besonders wenn mehrere Levels inkrementeller Backups verwendet werden). Es ist sehr einfach, solche Inhaltsverzeichnisse während des Backups zu erzeugen.

Wenn Sie mit `tar` oder `cpio` arbeiten, können Sie die Option `-v` für die Generierung der Inhaltsverzeichnisse verwenden:

```
$ today='date +%d%b%Y'
$ tar -cv /home > /backup/home_full_$today.TOC
                    oder
$ tar -cv /home | tee /backup/home_full_$today.TOC
```

Beide `tar`-Befehle archivieren den Inhalt von */home* und generieren gleichzeitig eine Liste der Dateinamen (im Stil langer Verzeichnis-Listings), die in einer Datei wie */backup/home_full_21mar1995.TOC* gespeichert wird. Der zweite Befehl gibt die gleiche Ausgabe auch auf dem Bildschirm aus.

`cpio` sendet die Dateiliste an Standardfehler, muss also etwas anders behandelt werden:

```
$ toc='date +/backup/home_full_%d%b%y.TOC'
$ find /home -print |  cpio -ov > /dev/rmt0 2> $toc
```

Wenn Sie die C-Shell verwenden wollen, sehen die Befehle etwas anders aus:

```
% set toc='date +/backup/home_full_%d%b%y.TOC'
% (find /home -print | cpio -ov > /dev/rmt0) >& $toc
```

Die mit solchen `cpio`-Befehlen generierten Dateilisten enthalten nur die Pfadnamen der Dateien im Archiv. Wenn Sie detailliertere Listings möchten, können Sie diese mit einem zweiten `cpio`-Befehl oder mit einer etwas komplexeren Pipe erzeugen:

```
$ cpio -itv < /dev/rmt0 > $toc
$ find /home | cpio -o | tee /dev/rmt0 | cpio -t -i -v > $toc
```

Der erste Befehl gibt die Dateien des Archivs aus. Der zweite Befehl verhindert, dass das Backup-Band erneut gelesen werden muss, indem er auf den `find`-Befehl zurückgreift, um eine Liste der Dateien zu erzeugen, die dann von `cpio` in einem Archiv abgelegt werden.

Dieses Archiv wird dann an ein Bandlaufwerk und einen weiteren cpio-Befehl gesendet. Dieser führt den Inhalt des Archivs auf und schreibt ihn in die angegebene Datei.

Die Generierung eines Inhaltsverzeichnisses eines dump-Bandes benötigt einen zusätzlichen restore-Befehl. Zum Beispiel sehen Sie nachfolgend ein Skript, das ein Backup mit dump und dann ein Inhaltsverzeichnis mit restore anlegt:

```
#!/bin/csh
# bkup+toc - Dump durchführen, Band prüfen und Inhaltsverzeichnis anlegen
# $1 = Dateisystem
# $2 = Dump-Level (voreingestellt 0)
#
if ($#argv < 1) then
  echo "do_backup: Dateisystem [Dump-Level]"
  exit 1
endif

set lev=0
if ("$2" != "") set lev=$2
dump -${lev} -u -f /dev/rmt1 $1
if ($status) then
  echo "do_backup: Dump fehlgeschlagen"
  exit 1
endif
restore -t -v -f /dev/rmt1 > /backup/`date +$1:t_%m-%d-%Y.$lev`
```

Das Skript führt den dump-Befehl auf dem Dateisystem aus, das als erstes Argument übergeben wurde. Die Sicherung erfolgt mit dem Backup-Level, der als zweites Argument übergeben wurde (oder standardmäßig mit Level 0). Beendet dump seine Arbeit fehlerfrei, wird der restore-Befehl verwendet, um das Backup zu überprüfen. Gleichzeitig wird der Inhalt des Bandes in eine Datei geschrieben, deren Name sich aus den Angaben zu Dateisystem, Monat, Tag und Jahr der Sicherung zusammensetzt. Als Dateierweiterung wird der Backup-Level verwendet. *chem_06-24-2001.2* wäre also der Dateiname eines Level-2-Backups des */chem*-Dateisystems, das am 24. Juni 2001 angelegt wurde.

Auf einem HP-UX-System können Sie den folgenden frecover-Befehl verwenden, um ein Inhaltsverzeichnis anzulegen:

```
# frecover -r -Nv -f /dev/rmt/0m > $toc
```

Netzwerk-Backup-Systeme

Bisher haben wir uns nur mit Backup- und Restore-Operationen der Platten eines lokalen Computers beschäftigt. Viele Organisationen benötigen aber einen einheitlicheren und umfassenderen Ansatz, um ihren Backup-Bedarf decken zu können. In diesem Abschnitt wollen wir verschiedene Lösungen betrachten, die für dieses Problem zur Verfügung stehen.

Entfernte Backups und Restores

Die einfachste Möglichkeit, die Grenze einfacher Ein-System-Backups hinter sich zu lassen, sind entfernte Backups und Restores. Es ist ein durchaus gängiger Wunsch, Backups über das Netzwerk auszuführen. Die Gründe dafür sind recht unterschiedlich: Ihr System besitzt möglicherweise gar kein Bandlaufwerk. Heutzutage werden nicht mehr alle Systeme standardmäßig mit einem Bandlaufwerk ausgeliefert, auf einem anderen System könnte ein besseres (d. h. schnelleres, mit höherer Kapazität ausgestattetes) Laufwerk vorhanden sein usw.

Die meisten Versionen von `dump` und `restore` können netzwerkbasierte Operationen durchführen (Tru64 verlangt die Verwendung der separaten Befehle `rdump` und `rrestore`). Man erreicht dies durch Angabe eines Gerätenamens der Form *host:lokales_gerät* als Argument für die Option `-f`. Dem Hostnamen kann optional auch ein Benutzername und ein at-Zeichen vorangestellt sein. Die Angabe `-f chavez@hamlet:/dev/rmt1` führt die Operation also auf dem Gerät */dev/rmt1* auf Host *hamlet* unter dem Benutzer *chavez* durch.

Diese Fähigkeit nutzt die gleichen Netzwerkdienste wie die `rsh`- und `rcp`-Befehle. Einrichtungen zum Erstellen entfernter Backups sind vom Daemon */usr/sbin/rmt* (häufig auf */etc/rmt* gelinkt)[14] abhängig. Um den Zugriff auf ein entferntes System zu ermöglichen, muss die Datei *.rhosts* in dessen root-Verzeichnis enthalten sein, die zumindest den Namen des (lokalen) Hosts enthalten muss, von dem die Daten kommen. Diese Datei muss *root* gehören und der Modus darf keinen Zugriff durch Gruppen oder andere Benutzer ermöglichen (z. B. 400). Dieser Mechanismus hat die für ihn typischen negativen Auswirkungen auf die Sicherheit (siehe »Netzwerksicherheit« in Kapitel 7).

Einige `tar`-Versionen können auch die »remote tape facility« `rmt` verwenden.

Die HP-UX-Utilities `fbackup` und `frestore` akzeptieren entfernte Bandlaufwerke als Argumente für die normale `-f`-Option. Ein Beispiel:

```
# fbackup -0u -f backuphost:/dev/rmt/1m -i /chem
```

Amanda

Amanda steht für Advanced Maryland Automated Network Disk Archiver. Das System wurde an der University of Maryland entwickelt (James da Silva war der ursprüngliche Autor). Die Homepage des Projekts finden Sie unter *http://www.amanda.org*, wo man es auch kostenlos herunterladen kann. Dieser Abschnitt enthält eine Übersicht über Amanda. In Kapitel 4 von *Unix Backup and Recovery* finden Sie eine sehr ausführliche Betrachtung aller Amanda-Features (dieses Kapitel ist ebenfalls auf der Amanda-Homepage verfügbar).

14 Auf einigen wenigen älteren Systemen müssen Sie den Link selbst erzeugen.

Über Amanda

Amanda erlaubt es einem Netzwerk von Clients, Backups an einen einzelnen designierten Backup-Server zu senden. Das Paket fungiert dabei als Wrapper um echte Backup-Software wie GNU `tar` und `dump`. Die Sicherung von Dateien von Windows-Clients ist über Samba (`smbtar`) ebenfalls möglich. Es hat eine Reihe schöner Features:

- Es verwendet seine eigenen Netzwerkprotokolle und weist daher nicht die Sicherheitsprobleme auf, die dem `rmt`-Ansatz innewohnen.
- Es unterstützt viele gängige Bandgeräte und andere Backup-Geräte (inklusive Stackern und Jukeboxen).
- Es kann vollständige und inkrementelle Backups durchführen und den Backup-Level basierend auf vorgegebenen Konfigurationsparametern automatisch festlegen.
- Es kann die Hardware-Komprimierung nutzen oder Archive komprimieren (wenn keine Hardware-Komprimierung zur Verfügung steht), bevor sie auf Band (oder andere Medien) geschrieben werden. Die Software-Komprimierung kann entweder vom Hauptserver oder vom Clientsystem vorgenommen werden.
- Es bietet hervorragenden Schutz vor dem versehentlichen Überschreiben von Medien.
- Es kann Backup-Archive auf Festplatten zwischenspeichern, um den Schreibdurchsatz auf Band zu erhöhen und um sicherzustellen, dass die Daten gesichert wurden, falls es zu einem Laufwerksfehler kommen sollte. (Das Backup kann dann zu einem späteren Zeitpunkt auf das Backup-Medium geschrieben werden.)
- Neben seinem eigenen Authentifizierungsschema kann es die Kerberos-Authentifizierung verwenden. Die Kerberos-Verschlüsselung kann auch verwendet werden, um die Daten während der Netzwerkübertragung zu schützen.

Momentan besitzt Amanda aber auch eine Reihe lästiger Einschränkungen:

- Es kann ein Backup-Archiv nicht auf mehrere Bänder verteilen. Wird beim Schreiben des Backups eine Bandende-Markierung gefunden, beginnt es auf dem nächsten Band wieder von vorne.
- Es kann keine Backup-Archive anlegen, die größer sind als ein einzelnes Band. Das ist die Konsequenz aus der ersten Einschränkung.
- Es wird nur ein einziger Backup-Server unterstützt.

Wie Amanda arbeitet

Amanda verwendet eine Kombination aus vollständigen und inkrementellen Backups, um alle Daten, für die es verantwortlich ist, mit dem kleinstmöglichen täglichen Backup zu sichern. Sein Schema berechnet zuerst die Gesamtmenge der zu sichernden Daten. Es verwendet diesen Gesamtwert zusammen mit einer Reihe von vom Administrator definierten Parametern, um herauszufinden, was im aktuellen Durchlauf zu tun ist. Hier die Schlüssel-Parameter:

Anzahl der Durchläufe in einem Backup-Zyklus
: Bei einer Rate von einem Amanda-Lauf pro Tag entspricht dies der gewünschten Anzahl von Tagen zwischen vollständigen Backups.

Prozentsatz der Daten, die sich zwischen den Amanda-Läufen ändern
Bei einem Durchlauf pro Tag ist dies der Prozentsatz der sich täglich ändernden Daten.

Amanda verfolgt eine Doppel-Strategie: eine vollständige Sicherung der Daten innerhalb jedes Zyklus und gleichzeitige Sicherstellung, dass alle veränderten Daten zwischen den vollständigen Dumps gesichert werden. Die traditionelle Methode besteht darin, ein vollständiges Backup durchzuführen und zwischen diesen Tagen inkrementelle Backups anzulegen. Amanda arbeitet anders.

Bei jedem (nächtlichen) Durchlauf führt Amanda ein vollständiges Backup eines Teils der Daten durch, genauer gesagt, jenes Teils, der benötigt wird, um die vollständigen Daten in einem kompletten Backup-Zyklus zu sichern. Ist dieser Zyklus also beispielsweise 7 Tage lang (mit einem Durchlauf pro Tag), muss täglich 1/7 der Daten gesichert werden, um in 7 Tagen ein vollständiges Backup abschließen zu können. Neben diesem »partiellen« vollständigen Backup nimmt Amanda auch inkrementelle Backups aller Daten vor, die sich seit dem letzten vollständigen Backup verändert haben.

Abbildung 11-1 zeigt einen Amanda-Backup-Zyklus von 4 Tagen, wobei angenommen wird, dass sich täglich 15% der Daten ändern. Das oberste Kästchen in der Abbildung repräsentiert den vollständigen Datenbestand, für den Amanda verantwortlich ist. Wir haben ihn in vier Segmente unterteilt, um die Teile darzustellen, die jeweils einem vollständigen Backup unterzogen werden.

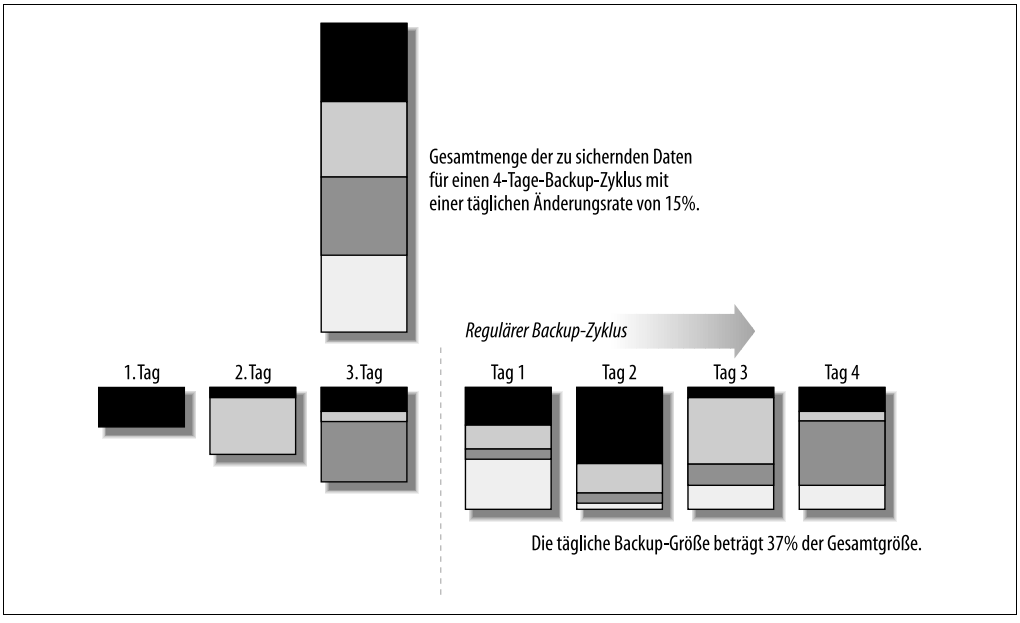


Abbildung 11-1: Das Amanda-Backup-Schema

Die Inhalte der nächtlichen Backups sind am unteren Rand der Abbildung zu sehen. Die ersten drei Tage stellen eine Startphase dar. In der ersten Nacht wird ein Viertel der Daten

vollständig gesichert. In der zweiten Nacht wird das zweite Viertel gesichert. Dazu kommen die 15% der Daten der vorigen Nacht, die sich während des zweiten Tages geändert haben. Am dritten Tag wird das dritte Viertel der Gesamtdatenmenge vollständig gesichert, ebenso wie die 15% des Backups des zweiten Tages. Zusätzlich werden 15% des in der ersten Nacht gesicherten Teils bis zum vollständigen Backup in den dazwischen liegenden Nächten gesichert. Mit anderen Worten: 30% dieses Viertels der Gesamtdaten.

Am vierten Tag tritt der normale Zeitplan in Kraft. Jede Nacht wird ein Viertel der gesamten Datenmenge vollständig gesichert, und inkrementelle Sicherungen aller anderen Viertel werden zu gegebener Zeit entsprechend dem letzten vollständigen Backup durchgeführt.

Dieses Beispiel verwendet nur inkrementelle Backups der ersten Ebene. In der Praxis verwendet Amanda mehrere Level inkrementeller Backups, um die Anforderungen an den Backup-Speicherplatz zu minimieren.

Um Dateien von einem Amanda-Backup wieder einzuspielen, benötigen Sie alle Medien eines vollständigen Zyklus.

Betrachten wir nun ein numerisches Beispiel. Nehmen wir an, wir besitzen 100 GB an Daten, die gesichert werden müssen. Tabelle 11-3 zeigt vier Amanda-Backup-Zeitpläne, die auf unterschiedlich langen Zyklen und täglichen prozentualen Veränderungen basieren.

Tabelle 11-3: Beispiele für Amanda-Backup-Größen (Gesamtdatenmenge=100 GB)

	3-Tage-Zyklus 10% Änderung	5-Tage-Zyklus 10% Änderung	7-Tage-Zyklus 10% Änderung	7-Tage-Zyklus 15% Änderung
Vollständiger Teil	33.3	20.0	14.3	14.3
Vortag	3.4	2.0	1.4	2.2
Vor-Vortag	6.8	4.0	2.8	4.4
Vor-Vor-Vortag		6.0	4.2	6.6
Vor-Vor-Vor-Vortag		8.0	5.6	8.8
Vor-Vor-Vor-Vor-Vortag			7.0	11.0
Vor-Vor-Vor-Vor-Vor-Vortag			8.4	13.2
Tägliche Menge (GB)	43.5	40.0	43.7	60.5

Die Spalten der Tabelle verdeutlichen die Datenmengen, die für das tägliche Backup notwendig sind. Unterteilt ist das Ganze in den Teil mit dem vollständigen Backup und den inkrementellen Daten jedes vorangegangenen vollständigen Backups innerhalb des Zyklus.

Beachten Sie, dass Amanda die Menge der zu sichernden Daten bei jedem Durchlauf neu berechnet, d.h., das Ganze ist nicht so statisch, wie es das Beispiel vielleicht vermittelt. Dennoch liefern die Beispiele ein allgemeines Bild davon, wie die Einrichtung arbeitet.

Im nächsten Abschnitt gehen wir etwas formeller darauf ein, wie die Größe des Backups vom Backup-Zyklus abhängt. Wir stellen auch einige Formeln vor, die zur Berechnung des passenden Backup-Zyklus unter bestimmten Bedingungen herangezogen werden können.

Bestimmung der täglichen Änderungsrate

Sie können den `find`-Befehl verwenden, um die tägliche Änderungsrate zu bestimmen:

```
$ find Verzeichnis -newer /var/adm/gestern -ls | \
  awk '{sum+=$7}; END {print "Differenz =",sum}'
```

Wiederholen Sie den Befehl, bis alle zu sichernden Daten berücksichtigt sind. Verwenden Sie `touch`, um die Zeit der Datei */var/adm/gestern* zu aktualisieren, nachdem alle `find`-Befehle ausgeführt wurden.

Teilen Sie diesen Wert dann durch den insgesamt verwendeten Speicherplatz (den Ihnen z.B. die `df`-Ausgabe liefern kann). Wiederholen Sie diesen Prozess mehrere Tage oder Wochen, um einen Durchschnittswert zu ermitteln.

Die Mathematik

Als Nächstes wollen wir uns einige Formeln ansehen, die zur Berechnung der Ausgangsparameter von Amanda genutzt (und mit der Zeit an die tatsächlichen Gegebenheiten angepasst) werden können. Wenn Sie diese Art der mathematischen Analyse nicht interessiert, können Sie diesen Abschnitt überspringen.

Wir verwenden die folgenden Variablen:

T= gesamte Datenmenge
p = prozentuale Veränderung zwischen den Durchläufen (in dezimaler Form, d. h. zum Beispiel 12 %=0.12)
n = Zahl der Durchläufe eines vollständigen Zyklus (häufig Tage)
S = Menge der Daten, die bei jedem Durchlauf gesichert werden müssen (Tag)
F = Teil der Daten, die bei jedem Durchlauf gesichert werden müssen (Tag): S/T

Um zu berechnen, welche Datenmenge pro Durchlauf gesichert werden muss, verwenden Sie die folgende Formel für S:

$$S = \frac{T}{n} + \frac{Tp(n-1)}{2}$$

Wenn beispielsweise 70 GB an Daten, von denen sich 10% pro Tag ändern, in einem 7-Tage-Zyklus gesichert werden sollen, dann müssen jede Nacht 31 GB auf Band geschrieben werden ($70/7 + 70 \times 0{,}1 \times 6/2 = 10 + 42/2 = 10 + 21 = 31$). Wenn diese 31 GB die maximale Kapazität in der zur Verfügung stehenden Zeit überschreiten, müssen Sie die anderen Parameter entsprechend korrigieren (siehe unten).

Wenn Ihnen alternativ pro Durchlauf eine feste Backup-Kapazität zur Verfügung steht, können Sie die benötigte Zykluslänge ermitteln. Wie viel Kapazität Ihnen zur Verfügung steht, haben wir bereits in diesem Kapitel im Abschnitt über die Kapazitätsplanung erläutert.

Um *n* für eine gegebene nächtliche Kapazität zu ermitteln, verwenden Sie die folgende Formel:

$$n = \frac{x \pm \sqrt{x^2 - 2p}}{p}$$

wobei

$$x = \left(\frac{p}{2} + \frac{S}{T}\right)$$

Wir haben die Variable *x* eingeführt, um die Formel für *n* zu vereinfachen. Nehmen wir an, Sie besitzen für das gleiche Szenario (eine gesamte Datenmenge von 70 GB bei einer täglichen Veränderung von 10%) eine Kapazität von 40 GB. Dann ist $x = 0{,}1/2 + 40/70 = 0{,}05 + 0{,}57 = 0{,}62$. Wir können nun *n* berechnen: $(0{,}62 \pm \sqrt{0{,}38 - 0{,}2}) / 0{,}1 = (0{,}62 \pm \sqrt{0{,}18}) / 0{,}1 = (0{,}62 \pm 0{,}42) / 0{,}1 = 6{,}2 \pm 4{,}2$.

Diese Berechnung liefert (auf ganze Zahlen gerundet) die Werte 2 und 11. Wir können also jede Nacht vollständige Backups von ungefähr der Hälfte der Daten anlegen oder einen wesentlich längeren Zyklus von 11 Tagen verwenden. Beachten Sie, dass diese Werte die verfügbare Kapazität bis an ihre Grenzen ausreizen.

Nehmen wir nun an, dass wir nur über eine nächtliche Kapazität von 20 GB für das gleiche Szenario verfügen (eine gesamte Datenmenge von 70 GB bei 10% täglicher Änderung). Dann ist $x = 0{,}1/2 + 40/70 = 0{,}05 + 0{,}29 = 0{,}34$ und *n*: $(0{,}34 \pm \sqrt{0{,}12 - 0{,}2}) / 0{,}1$. Der die Quadratwurzel enthaltende Ausdruck ist nun imaginär (weil 0.12–0.20 negativ ist), was bedeutet, dass die angedachte Konfiguration in der Praxis nicht funktioniert.[15] Die verfügbare Kapazität ist einfach zu gering.

Ganz allgemein können Sie die minimale Kapazität pro Durchlauf für eine gegebene prozentuale Veränderung pro Durchlauf (*p*) mit der folgenden Formel berechnen (die *F* als Teil der zu sichernden Gesamtdatenmenge einführt):

$$F_{\text{min für festes } p} = 2\sqrt{\frac{p}{2}} - \frac{p}{2} \quad (\text{wobei } F = \frac{S}{T}); \Rightarrow S_{min} = F_{min} \times T$$

F steht dabei für den Teil der Daten, der bei jedem Durchlauf gesichert werden muss, damit das System Erfolg haben kann. In unserem Fall einer Änderungsrate von 10% ist $F = 2 \times \sqrt{0{,}1/2} - (0{,}1/2) = 2 \times \sqrt{0{,}05} - 0{,}05 = 2 \times 0{,}22 - 0{,}05 = 0{,}44 - 0{,}05 = 0{,}39 \approx 40\%$. Beachten Sie, dass dieser Ausdruck von *T* (der zu sichernden Gesamtdatenmenge) unabhängig ist. Wann immer sich die Daten pro Durchlauf um etwa 10% verändern, müssen Sie in der Lage

15 Mathematisch gibt es keine reellen Lösungen für die zugrunde liegende quadratische Gleichung.

sein, zumindest 40% der Gesamtdatenmenge bei jedem Durchlauf zu sichern. In unserem Fall entspricht das einer minimalen Kapazität von $0{,}4 \times 70 = 29$ GB pro Nacht.

Alternativ können Sie den Laufzyklus *n* berechnen, der notwendig ist, um *F* (und damit *S*) für ein gegebenes *p* zu berechnen. Hier die Formel:[16]

$$n_{min\,S} = \sqrt{\frac{2}{p}}$$

In unserem Fall beträgt der Zyklus, der die zu sichernde Datenmenge minimiert, $\sqrt{2/0{,}01} = \sqrt{20} = 4{,}47 \approx 5$ Tage. Auch dieser Wert ist wieder unabhängig von der Menge der zu sichernden Daten. In unserem Fall, bei dem sich die Daten im Bereich von etwa 10% pro Tag ändern, minimiert ein Zyklus von 5 Tagen die Menge an Daten, die jede Nacht zu sichern sind. Das ist die effektivste Zykluslänge für die minimale nächtliche Backup-Kapazität.

Sowohl die minimale Zyklusdauer als auch die Datenmenge der pro Durchlauf zu sichernden Daten werden also nur anhand der Änderungsrate bestimmt. Die tatsächliche Backup-Größe für die Gesamtmenge der zu sichernden Daten lässt sich daraus einfach herleiten. Daher ist eine genaue Bestimmung von *p* für die rationale Planung von entscheidender Bedeutung.

Diese Betrachtung ignoriert die Komprimierung. Wenn Ihr Bandlaufwerk Daten komprimieren kann oder wenn Sie entscheiden, die Daten vor dem Schreiben auf Band durch Software komprimieren zu lassen, müssen Sie den erwarteten Komprimierungsfaktor in Ihre Berechnungen einfließen lassen.

Amanda konfigurieren

Die Kompilierung und Installation von Amanda ist recht einfach. Da der Prozess gut dokumentiert ist, werden wir hier nicht weiter auf ihn eingehen.

Das Amanda-System umfasst die folgenden Komponenten:

- Client-Programme, von denen `amandad` das wichtigste ist. Dieser Daemon kommuniziert während der Backup-Läufe mit dem Amanda-Server und ruft bei Bedarf weitere Client-Programme auf: `selfcheck` (Verifikation der lokalen Amanda-Konfiguration), `sendsize` (geschätzte Backup-Größe), `sendbackup` (Durchführung von Backup-Operationen) und `amcheck` (Verifikation des Amanda-Setups). Diese Programme sind Teil des Amanda-Client-Systems. Auf dem Amanda-Server liegen diese Programme zusammen mit den anderen Hilfsprogrammen des Pakets in */usr/local/lib/amanda* oder */usr/lib/amanda*.
- Server-Programme zur Durchführung der verschiedenen Phasen der eigentlichen Backup-Operationen. Das `amdump`-Programm stößt einen Amanda-Lauf an und wird üblicherweise in regelmäßigen Abständen von `cron` ausgeführt. Es kontrolliert eine Reihe weiterer Programme, darunter `planner` (bestimmt, was gesichert werden muss),

16 Mathematisch ist dies der Wert von *n* an der Stelle $\partial F / \partial n = 0$. In diesem spezifischen Fall ist die mathematische Region um das Minimum sehr flach.

`driver` (Schnittstelle zum Gerät), `dumper` (kommuniziert mit `amandad`-Prozessen auf Clients), `taper` (schreibt Daten auf das Medium) und `amreport` (erzeugt einen Bericht zu einem Amanda-Lauf).

- Administrative Utilities zur Erledigung entsprechender Aufgaben. Hierzu zählen `amcheck` (prüft, ob die Amanda-Konfiguration gültig und ob das System betriebsbereit ist), `amlabel` (bereitet Medien für den Einsatz mit Amanda vor), `amcleanup` (räumt nach einem abgebrochenen Durchlauf oder einem Systemabsturz auf), `amflush` (erzwingt das Schreiben von Daten aus dem Zwischenspeicher auf das Backup-Medium) und `amadmin` (führt verschiedene administrative Funktionen durch).
- Die Konfigurationsdateien, die Amanda-Operationen beschreiben. Diese enthalten Angaben darüber, was wie oft gesichert werden soll, aber auch die Lage und die Eigenschaften des Bandlaufwerks. Diese Dateien heißen *amanda.conf* und *disklist* und liegen in einem Unterverzeichnis des Amanda-Hauptverzeichnisses (eigentlich */usr/local/etc/amanda*, aber auch */etc/amanda*, wenn das Paket schon vorinstalliert ist). Ein typischer Name ist *Daily*. Jedes Unterverzeichnis entspricht einer Amanda-»Konfiguration«, d.h. einem separaten Satz von Einstellungen und Optionen, die über den Verzeichnisnamen angesprochen werden.
- Das `amrestore`-Utility, mit dessen Hilfe Daten von Amanda-Backups wieder eingespielt werden können. Darüber hinaus unterstützt das `amrecover`-Utility die interaktive Wiederherstellung von Dateien. Es ist von Daemons abhängig, um seine Arbeit erledigen zu können: `amindexd` und `amidxtaped`.

Einrichten eines Amanda-Clients. Sobald Sie die Amanda-Software auf einem Client installiert haben, sind noch einige zusätzliche Schritte notwendig. Zuerst müssen Sie zusätzliche Einträge in die Dateien */etc/inetd.conf* und */etc/services* eintragen, um die Unterstützung der Amanda-Netzwerkdienste zu aktivieren:

/etc/services:
```
amanda    10080/udp
```

/etc/inetd.conf:
```
amanda  dgram   udp   wait   amanda   /Pfad/amandad   amandad
```

Der Amanda-Daemon läuft in diesem Beispiel unter dem Benutzer *amanda*. Sie müssen natürlich den Benutzernamen eintragen, den Sie bei der Installation der Amanda-Software festgelegt haben.

Darüber hinaus müssen Sie sicherstellen, dass alle zu sichernden Daten vom Amanda-Benutzer und der -Gruppe gelesen werden können. Auch die Datei */etc/dumpdates* muss existieren und von der Amanda-Gruppe gelesen werden können.

Abschließend muss das von `amandad` verwendete Authentifizierungsschema festgelegt werden. Dies geschieht üblicherweise bei der Kompilierung. Sie können die normale *.rhosts*-basierte Authentifizierung, die Kerberos-Authentifizierung (siehe unten) oder eine separate *.amandahosts* (der Standardmechanismus) verwenden. Die Datei *.amandahosts* ähnelt der *.rhosts*-Datei, gilt aber nur für die Amanda-Einrichtung und besitzt daher nicht das

gleiche Risiko. Vollständige Informationen zu den Authentifizierungsoptionen finden Sie in der Amanda-Dokumentation.

Wahl eines Amanda-Servers. Die Wahl eines geeigneten Systems für den Amanda-Server ist für einen guten Durchsatz von entscheidender Bedeutung. Sie sollten dabei folgende Dinge berücksichtigen:

- Das System sollte die bestmöglichen Bandlaufwerke (oder andere Backup-Geräte) besitzen.
- Das System sollte über eine ausreichende Netzwerk-Bandbreite für den zu erwartenden Datenfluss verfügen.
- Das System sollte ausreichend Plattenspeicher als Zwischenspeicher zur Verfügung stellen. Ein guter Wert ist das doppelte der größtmöglichen Datenmenge eines Durchlaufs.
- Wenn der Server eine Software-Komprimierung der Daten vornimmt, ist eine schnelle CPU notwendig.
- Große Mengen Arbeitsspeicher wirken sich kaum auf den Backup-Durchsatz aus. Es gibt also keinen Grund, das System mit zu viel Arbeitsspeicher zu versehen.

Einrichten des Amanda-Servers. Nach der Installation der Software sind noch verschiedene Schritte notwendig, um den Amanda-Server zu konfigurieren. Zuerst müssen Sie die gleichen Einträge wie bei den Amanda-Clients auch in den Netzwerk-Konfigurationsdateien des Servers vornehmen:

/etc/services:
```
amanda          10080/udp
amandaidx       10082/tcp
amidxtape       10083/tcp
```

/etc/inetd.conf:
```
amandaidx  stream  tcp  nowait  amanda /Pfad/amindexd   amindexd
amidxtape  stream  tcp  nowait  amanda /Pfad/amidxtaped amidxtaped
```

Als Nächstes müssen Sie Amanda konfigurieren, indem Sie die notwendigen Konfigurationsdateien erstellen. Legen Sie bei Bedarf ein neues Unterverzeichnis *etc/amanda* im obersten Amanda-Verzeichnis an (z.B. */usr/local* oder */*). Wir verwenden *Daily* für unser Beispiel. Dann legen wir die Konfigurationsdateien *amanda.conf* und *disklist* in diesem Unterverzeichnis an und modifizieren sie entsprechend unseren Anforderungen (das Amanda-Paket enthält Beispieldaten, die als Ausgangspunkt dienen können).

Wir beginnen mit *amanda.conf* und sehen uns ihren Inhalt in Gruppen verwandter Einträge an. Wir sehen uns die beiliegende *amanda.conf*-Beispieldatei an.

Die ersten Einträge in der Datei legen normalerweise Informationen zur lokalen Site und der Lage wichtiger Dateien fest:

```
org "ahania.com"                Name der Organisation in Reports.
mailto "amanda-rep"             Reports gehen per Mail an diesen Benutzer.
```

```
dumpuser "amanda"               Amanda-Benutzer-Account.
printer "tlabels"               Drucker für Band-Label.
logdir "/var/log/amanda"        Log-Dateien finden sich hier.
indexdir "/var/adm/amindex"     Indexdaten zum Backup-Satz finden sich hier.
```

Die nächsten paar Einträge legen grundsätzliche Parameter für die Backup-Prozedur fest:

```
# Elementare Parameter
dumpcycle 7 days      Länge des Backup-Zyklus (standardmäßig 10 Tage).
runspercycle 5        Amanda-Läufe pro Tag (wenn < 1/Tag).

# Netzwerkrelevante Ressourcen-Einstellungen
netusage 400 kps      Maximale Netzwerk-Bandbreite (standardmäßig 300).
inparallel 20         Max. simultane Backups (standardmäßig 10).
ctimeout 120          Timeout-Zeit für den Client (standardmäßig 30 Sekunden).

# Bump-Parameter für inkrementelle Level
bumpsize 20 mb        Min. Einsparung für Level 2 (standardmäßig 10).
bumpdays 1            Notwendige Anzahl von Tagen in jedem Level (standardmäßig 2).
bumpmult 2            Multipliziere Bump-Größe mit diesem Wert, um zum jeweils nächsten
                      inkrementellen Level zu kommen (standardmäßig 1,5).
```

Die incremental bump level Parameter legen fest, wann Amanda das Level für inkrementelle Backups erhöhen soll, damit die Größe des Backup-Satzes kleiner wird. Bei unseren Einstellungen wechselt Amanda von Level 1- zu Level 2-Inkrementals, sobald es mindestens 20 MB an Speicherplatz spart. Der Multiplikationsfaktor hat den Effekt, dass zusätzliche Einsparungen notwendig sind, um zum jeweils höheren Level zu wechseln. Der Grenzwert für jeden Level berechnet sich aus diesem Faktor mal der Einsparung des vorherigen Levels, d.h. in unserem Beispiel 40 für Level 2 nach 3, 80 für Level 3 nach 4 und so weiter. Diese Strategie wird verwendet, um sicherzustellen, dass die zusätzliche Komplexität mehrerer Level inkrementeller Backups auch deutliche Einsparungen für die Größe des Backup-Satzes bringen.

Die folgenden Einträge enthalten Informationen über das zu verwendende Bandlaufwerk und das Medium:

```
# Anzahl der Bänder             Geben Sie zumindest die Anzahl der Bänder an, die für einen
                                vollständigen Zyklus notwendig sind,
tapecycle 25                    sowie einige zusätzliche freie (standardmäßig 15).
labelstr "Daily[0-9][0-9]*"     Format der Tabellen-Label (regulärer Ausdruck).

tapedev "/dev/rmt/0"
tapetype "DLT"

#changerdev "/dev/whatever"
#tpchanger "script-path"          Skript zum Wechseln des nächsten Bandes.
#runtapes 4                      Maximale Anzahl zu nutzender Bänder.
```

Die beiden ersten Einträge legen die Anzahl der zu verwendenden Bänder sowie das Muster für deren elektronische Label fest. Beachten Sie, dass die Bänder vor dem Einsatz mit `amlabel` vorbereitet werden müssen (was nachfolgend noch erläutert wird).

Die darauf folgenden beiden Einträge geben die Lage des Bandlaufwerks und dessen Typ an. Die letzten drei Einträge werden von Bandwechslern verwendet und sind in diesem Beispiel auskommentiert. Es darf nur *tapedev* oder *tpchanger* verwendet werden.

Bandtypen werden an anderer Stelle der Konfigurationsdatei mit Schablonen wie der folgenden definiert:

```
define tapetype DLT {
    comment "DLT mit 10-GB-Bändern"
    length 12500 mb       Bandkapazität (berücksichtigt die Komprimierung).
    speed 1536 kps        Laufwerksgeschwindigkeit.
    lbl-templ "Datei"     PostScript-Template für gedruckte Label.
}
```

Die Beispiel-Konfigurationsdatei enthält viele definierte Bandtypen. Die *length*- und *speed*-Parameter werden nur für Schätzungen verwendet (z. B. wie viele Bänder benötigt werden). Bei der eigentlichen Datenübertragung auf Band schreibt Amanda so lange weiter, bis die Bandende-Markierung erreicht wird.

Der folgende Eintrag und die *holdingdisk*-Schablone definieren den Zwischenspeicher auf der Festplatte:

```
# Ist das Medium nicht verfügbar, wird dieser Prozentsatz des Zwischenspeichers
# für inkrementelle Backups im degraded-mode reserviert.
reserve 50                Standardmäßig 100%.

holdingdisk amhold0 {     Name ist amhold0.
   comment "Primärer Zwischenspeicher"
   directory "/scratch/amanda"
# Zu verwendender (+) oder zu reservierender Speicherplatz (-);
# 0=alles verwenden (Standard)
   use -2 Gb              Immer so viel Platz übrig lassen.
}
```

Sie können mehr als einen Festplatten-Zwischenspeicher definieren.

Die letzte in der Konfigurationsdatei vorzunehmende Aufgabe ist die Definition der verschiedenen Dump-Typen: allgemeine Backup-Operationen mit bestimmten Eigenschaften (aber unabhängig von den zu sichernden Daten). Hier ein Beispiel für den *normal*en Backup-Typ (Sie können beliebige Namen auswählen):

```
define dumptype normal {
   comment "Normales Backup"
   holdingdisk yes      Auf Festplatte zwischenspeichern.
   index yes            Index-Informationen zum Inhalt pflegen.
   program "DUMP"       Backup-Befehl.
   priority medium      Relative Backup-Priorität festlegen.
#  24-Stunden-Format ohne Interpunktionszeichen
   starttime 2000       Keine Backups vor diesem Zeitpunkt (hier 8 Uhr abends).
}
```

Dieser Dump-Typ speichert die Daten auf einer Platte zwischen, erzeugt einen Index für den Inhalt des Backup-Satzes (was die interaktive Wiederherstellung ermöglicht) und ver-

wendet das dump-Programm für die eigentliche Sicherung. Im Vergleich zu anderen Backups wird es mit einer mittleren Priorität ausgeführt (die möglichen Werte sind *low* (0), *medium* (1), *high* (2) sowie ein beliebiger Integerwert, wobei höhere Werte für ein früheres Backup stehen). Backups, die diese Methode verwenden, beginnen mit der Sicherung nicht vor acht Uhr abends, unabhängig davon, wann der amdump-Befehl ausgeführt wurde.

Amanda stellt mehrere vordefinierte Dump-Typen in der Beispieldatei *amanda.conf* zur Verfügung, die einfach verwendet oder an die eigenen Bedürfnisse angepasst werden können.

Hier einige weitere Parameter, die bei der Definition von Dump-Typen nützlich sind:

```
program "GNUTAR"          Verwendet GNU tar für Backups.
                          Diesen Wert müssen Sie auch bei Samba-Backups angeben.
exclude ".exclude"        GNU-tar-Ausschluss-Datei (im obersten Verzeichnis
                          des zu sichernden Dateisystems).
compress server "fast"    Verwende Software-Komprimierung auf dem Server
                          mit der schnellsten Komprimierungsmethode. Andere
                          Schlüsselwörter sind »client« und »best«.
auth "krb4"               Verwende Kerberos 4-Benutzer-Authentifizierung.
kencrypt yes              Verschlüssele übertragene Daten.
ignore yes                Diesen Backup-Typ nicht ausführen.
```

Die Amanda-Konfigurationsdatei *disklist* bestimmt die Dateisysteme, die es eigentlich zu sichern gilt. Hier einige Beispieleinträge:

```
# Host     Partition      Dump-Typ    Spindel-Parameter
hamlet      sd1a           normal        -1
hamlet      sd2a           normal        -1
dalton      /chem          srv_comp      -1
leda        //leda/e       samba         -1   # Windows 2000-System
astarte     /data1         normal         1
astarte     /data2         normal         1
astarte     /home          normal         2
```

Die einzelnen Spalten dieser Datei enthalten den Hostnamen, die Plattenpartition (als Datei in */dev*, vollständiger Name der Gerätedatei oder Mountpunkt), den Dump-Typ und einen Spindel-Parameter. Dieser letzte Parameter kontrolliert, welche Backups auf einem Host zur gleichen Zeit durchgeführt werden können. Der Wert -1 gibt an, dass dieser Parameter ignoriert werden soll. Andere Werte definieren Backup-Gruppen innerhalb eines Hosts. Beim Host *astarte* muss zum Beispiel das */home*-Dateisystem separat von den beiden anderen gesichert werden (die wiederum gleichzeitig gesichert werden können, wenn Amanda es wünscht).

Es sind noch einige letzte Schritte notwendig, um das Setup des Amanda-Servers abzuschließen:

- Vorbereiten der Medien mit dem amlabel-Befehl. Der folgende Befehl bereitet beispielsweise ein mit »DAILY05« bezeichnetes Band für die Amanda-Konfiguration *Daily* vor:

  ```
  $ amlabel Daily DAILY05
  ```

  Ähnlich bereitet der folgende Befehl das Band in Slot 5 des entsprechenden Bandlaufwerks als »CHEM101« für die *Chem*-Konfiguration vor:

```
$ amlabel Chem CHEM101 slot 5
```

- Verwenden Sie den `amcheck`-Befehl, um die Amanda-Konfiguration zu überprüfen.
- Legen Sie einen `cron`-Job für den Amanda-Benutzer an, der den `amdump`-Befehl regelmäßig (z. B. jede Nacht) ausführt. Der Befehl verlangt die gewünschte Konfiguration als Argument.

Amanda erwartet zu Beginn des Backup-Prozesses das richtige Band im Bandlaufwerk. Sie können bestimmen, welches Band die *Daily*-Konfiguration als Nächstes braucht, indem Sie den folgenden Befehl ausführen:

```
# amadmin Daily tape
```

Das Amanda-System verlangt eine fortlaufende Administration, inklusive Tuning und einiger Aufräumarbeiten. Letzteres geschieht über die Befehle `amflush` und `amcleanup`. `amflush` wird verwendet, um die Daten des Platten-Zwischenspeichers auf das Backup-Medium zu schreiben. Der Befehl wird normalerweise gebraucht, wenn während eines Sicherungslaufs ein Fehler auf dem Medium auftritt. In solchen Fällen werden die Backup-Daten dennoch in den Zwischenspeicher geschrieben. Der Befehl `amcleanup` muss ausgeführt werden, nachdem ein Amanda-Lauf abgebrochen oder ein Systemabsturz aufgetreten ist.

Schließlich können Sie eine Amanda-Konfiguration temporär deaktivieren, indem Sie eine Datei namens *hold* im entsprechenden Unterverzeichnis anlegen. Solange diese Datei existiert, legt das Amanda-System eine Pause ein. Sie können das nutzen, um die Konfigurationsinformationen zu sichern, falls ein Gerät einen Fehler aufweist oder falls ein Gerät kurzfristig für andere Aufgaben benötigt wird.

Amanda-Reports und -Logs

Das Amanda-System erzeugt einen Bericht, einen Report, für jeden Backup-Lauf und sendet diesen per E-Mail an den in der Konfigurationsdatei *amanda.conf* angegebenen Benutzer. Die Reports sind recht umfangreich und enthalten die folgenden Abschnitte:

- Datum und Uhrzeit des Dumps und die geschätzten Anforderungen an das Medium:

  ```
  These dumps were to tape DAILY05.
  Tonight's dumps should go onto one tape: DAILY05.
  ```

- Eine Zusammenfassung der Fehler und anderer Auffälligkeiten des letzten Durchlaufs:

  ```
  FAILURE AND STRANGE DUMP SUMMARY:
  dalton.ahania.com /chem lev 0 FAILED [request ... timed out.]
  ```

  Der Host *dalton* war ausgeschaltet, weshalb das Backup fehlgeschlagen ist.

- Statistiken zum Durchlauf, einschließlich der Datenmenge und der Schreibgeschwindigkeiten (die Ausgabe wurde gekürzt):

  ```
  STATISTICS:
                            Total       Full      Daily
                          --------  --------   --------
  Dump Time (hrs:min)         2:48       2:21       0:27
  Output Size (meg)         9344.3     7221.1     2123.2
  Original Size (meg)       9344.3     7221.1     2123.2
  Avg Compressed Size (%)     --         --         --
  ```

```
Tape Used (%)              93.4      72.2       21.2
Filesystems Dumped           10         2          8
Avg Dump Rate (k/s)      1032.1    1322.7      398.1
Avg Tp Write Rate (k/s)  1234.6    1556.2     1123.8
```

- Weitere Informationen zu Fehlern bzw. Auffälligkeiten, wenn diese verfügbar sind.
- Informationsmeldungen der verschiedenen von amdump aufgerufenen Unterprogramme:

```
NOTES:
   planner: Adding new disk hamlet.ahania.com:/sda2
   taper: tape DAILY05 9568563 kb fm 1 [OK]
```

- Eine zusammenfassende Tabelle zu den gesicherten Daten und damit zusammenhängende Informationen:

```
DUMP SUMMARY:
                         DUMPER STATS              TAPER STATS
HOST   DISK  L ORIG-KB OUT-KB COMP% MMM:SS  KB/s MMM:SS  KB/s
-------------------------------------------------------------
hamlet sd1a  1   28255  28255   --    2:36 180.3  0:21 1321.1
hamlet sd2a  0  466523 466523   --   36:51 211.1  5:33 1400.8
dalton /chem 1  FAILED---------------------------------------
ada    /home 1   39781  39781   --    5:16 125.7  0:29 1356.7
...
```

Sie sollten diese Reports regelmäßig prüfen, insbesondere die Abschnitte zu Fehlern und Performance.

Amanda kann auch Log-Dateien jedes Durchlaufs (*amdump.n* und *log.datum.n*) erzeugen, die sich im festgelegten Log-Verzeichnis befinden. Dabei handelt es sich um ausführlichere Versionen des E-Mail-Reports, die bei der Suche nach bestimmten Problemen hilfreich sein können.

Wiederherstellung von Dateien aus einem Amanda-Backup

Amanda stellt das interaktive amrecover-Utility zur Wiederherstellung von Dateien aus Amanda-Backups zur Verfügung. Hierzu müssen die Backup-Sätze indexiert (d.h. Sie müssen *index yes* setzen) und die beiden vorhin erwähnten Indexing-Daemons aktiv sein. Das Utility muss unter *root* vom richtigen Clientsystem aufgerufen werden.

Hier eine Beispiel-Sitzung:

```
# amrecover Daily
AMRECOVER Version 2.4.2. Contacting server on depot.ahania.com ...
...
Setting restore date to today (2001-08-12)
200 Working date set to 2001-08-14.
200 Config set to Daily.
200 Dump host set to astarte.ahania.com.
$CWD '/home/chavez/data' is on disk '/home' mounted at '/home'.
200 Disk set to /home.
amrecover> cd chavez/data
/home/chavez/data
amrecover> add jetfuel.jpg
Added /chavez/data/jetfuel.jpg
```

```
amrecover> extract
Extracting files using tape drive /dev/rmt0 on host depot...
The following tapes are needed: DAILY02
Restoring files into directory /home
Continue? [Y/n]: y
Load tape DAILY02 now
Continue? [Y/n]: y
warning: ./chavez: File exists
Warning: ./chavez/data: File exists
Set owner/mode for '.'? [yn]: n
amrecover> quit
```

In diesem Fall erinnert der amrecover-Befehl sehr stark an den Standard-restore-Befehl im interaktiven Modus.

Der amrestore-Befehl kann ebenfalls verwendet werden, um Daten aus einem Amanda-Backup wiederherzustellen. Details zu seinem Einsatz finden Sie in der entsprechenden Manpage und dem entsprechenden Abschnitt in *Unix Backup and Restore*.

Kommerzielle Backup-Pakete

Mehrere ausgezeichnete kommerzielle Backup-Einrichtungen stehen zur Verfügung. Eine aktuelle Liste der momentan verfügbaren Pakete finden Sie unter *http://www.storagemountain.com*. Wir wollen hier kein bestimmtes Paket betrachten, sondern vielmehr die wichtigen Features eines allgemein einsetzbaren Backup-Pakets zusammenfassen. Diese Liste können Sie dann als Kriterium für den Vergleich und die Evaluierung der Produkte heranziehen, die für Ihre Site interessant sein könnten.

Die folgenden Features sollten Sie von kommerziellen High-End-Backup-Paketen erwarten, die für mittelgroße bis große Netzwerke ausgelegt sind:

- Die Fähigkeit, Backup-Sätze als beliebige Listen von Dateien zu definieren, die vom jeweiligen Utility ganz nach Bedarf gesichert und wieder eingelesen werden können.
- Die Fähigkeit, die Eigenschaften und Daten zu definieren und zu speichern, aus denen Standard-Backup-Operationen bestehen.
- Eine Einrichtung für Listen auszuschließender Dateien. Mit dieser Einrichtung sollten Listen von Dateien und Verzeichnissen (mit Wildcard-Spezifikationen) erzeugt, gesichert und geladen werden können, die bei Backup-Operationen ausgeschlossen werden sollen.
- Ein automatisiertes Backup-Scheduling-System, das aus dem Backup-Utility selbst aufgerufen und gesteuert werden kann.
- Die Fähigkeit, Standardeinstellungen für Backup- und Restore-Operationen festzulegen.
- Die Fähigkeit, alle wichtigen Dateitypen (z.B. Gerätedateien, Sparse-Files) und Attribute (z.B. Zugriffskontroll-Listen) sichern zu können.
- Die Fähigkeit, offene Dateien sichern oder (je nach Wunsch) ohne Pause überspringen zu können.

- Die Fähigkeit, entfernte Backup- und Restore-Operationen definieren und initiieren zu können.
- Unterstützung mehrerer Backup-Server.
- Unterstützung von High-End-Backup-Geräten wie Stackern, Jukeboxen, Libraries und Silos.
- Unterstützung von RAID-fähigen Bandgeräten, bei denen mehrere physikalische Bänder über parallele Schreiboperationen zu einer einzigen logischen Hochleistungseinheit kombiniert werden.
- Unterstützung von nicht bandorientierten Backup-Geräten wie etwa Wechselplatten.
- Die Fähigkeit, mehrere Operationen gleichzeitig mit verschiedenen Bandgeräten durchzuführen.
- Unterstützung von Multiplex-Backups, bei denen mehrere Daten-Streams gleichzeitig auf einem Bandgerät gesichert werden.
- Die Fähigkeit der Clients, auf allen bei Ihrer Site installierten Betriebssystemen zu laufen.
- Kompatibilität mit Standard-Backup-Utilities, die für einige Sites sehr wichtig sein kann (so dass gesicherte Dateien auf jedem System wiederhergestellt werden können).
- Einrichtungen zur automatischen Archivierung inaktiver Dateien auf alternative Speichergeräte (zum Beispiel Jukeboxen oder optische Platten), um Plattenplatz zu sparen und die Backup-Anforderungen zu minimieren.
- Aufnahme irgendeiner Form von Datenbank-Manager, damit Sie (und die Backup-Software) Abfragen vornehmen können, um die Medien zu finden, die zur Wiederherstellung von Dateien benötigt werden.

Eine umfassendere Erläuterung der Features kommerzieller Backup-Pakete finden Sie in Kapitel 5 von *Unix Backup and Recovery*.

Backup und Restore der System-Dateisysteme

Dieser letzte Abschnitt behandelt die Sicherung und Wiederherstellung des Dateisystems, das das Betriebssystem selbst enthält. Wir besprechen auch den Fall einer fehlerhaften Systemfestplatte. Für das Recovery nach einer solchen Katastrophe hat sich in letzter Zeit der Begriff »Bare Metal Recovery« eingebürgert. *Unix Backup and Restore* enthält umfangreiche Kapitel, die diese Techniken und Prozeduren für verschiedene Unix-Varianten beschreiben.

Dateisysteme, die Betriebssystemdateien wie */* und */usr* enthalten, werfen einige Probleme auf, wenn Sicherungen verwendet werden sollen, um versehentlich gelöschte oder auf andere Weise verschollene Dateien wiederherzustellen. Wenn es sich bei der fraglichen Datei um eine nicht veränderte Systemdatei handelt, können Sie sie normalerweise von den Betriebssystemmedien wieder einspielen. Voraussetzung ist natürlich, dass Sie die Medien besitzen und unter normalen Bedingungen lesen können. Wenn eine dieser Bedin-

gungen nicht erfüllt ist, sollten Sie von Zeit zu Zeit ein vollständiges Backup der System-Dateisysteme anlegen.

Die von Ihnen modifizierten Dateien in den Systempartitionen sollten regelmäßig gesichert werden. In Kapitel 14 werden wir ein Skript vorstellen, das alle modifizierten Konfigurationsdateien und anderen Dateien des Benutzer-Dateisystems sichert. Die Sicherung erfolgt dabei regelmäßig und automatisch über die System-Backup-Prozeduren. Alternativ könnte das Skript sie direkt auf das Backup-Medium sichern (wenn das Archiv klein genug ist, sogar auf eine Diskette).

Wenn System-Dateisysteme vollständig wiederhergestellt werden müssen (üblicherweise auf Grund eines Hardwareproblems), kommen einige spezielle Erwägungen ins Spiel. Häufig gibt es zwei verschiedene Ansätze, die angewendet werden können:

- Neuinstallation von den Original-Installationsmedien, gefolgt von der Wiederherstellung der von Ihnen modifizierten Dateien. Bei diesem Ansatz könnte die Rekonfiguration einiger Subsysteme notwendig sein.
- Booten von einem alternativen Medium und Wiederherstellung der Dateisysteme von einem vollständigen Backup, das Sie angelegt haben.

Welche Alternative vorzuziehen ist, hängt stark von den Eigenschaften Ihres jeweiligen Systems ab: Wie viele Dateien wurden verändert und wie weit sind diese über die verschiedenen Dateisysteme verstreut, wie viele Geräte- und andere Konfigurationen müssen erneut durchgeführt werden und ähnliche Erwägungen. Wenn Sie mehrere Partitionen wiederherstellen müssen, ist es vielleicht günstiger, das Betriebssystem völlig neu aufzuspielen (es sei denn, es befinden sich nicht gesicherte Daten auf einer anderen Partition der gleichen Platte).

Wenn Sie sich für die zweite Vorgehensweise entscheiden, müssen Sie zuverlässige vollständige Backups des Systems anlegen, und zwar immer dann, wenn Sie signifikante Änderungen vorgenommen haben. Weil Sie in einem Notfall auf diese Bänder angewiesen sind, sollten die Backups immer überprüft oder sogar doppelt angelegt werden.

Auf jeden Fall müssen Sie zwischendurch auch Records der Plattenpartitionen, des assoziierten Dateisystem-Layouts und (wenn ein LVM aktiv ist) der Konfiguration der logischen Volumes zu Rate ziehen. Das ist von besonderer Bedeutung, wenn die Systemfestplatte beschädigt wurde und ersetzt werden muss, um das System in seiner vorherigen Konfiguration wiederherzustellen. Stellen Sie sicher, dass Sie entsprechende Aufzeichnungen von diesen Daten besitzen (siehe unten).

Nachfolgend eine allgemeine Prozedur zur Wiederherstellung eines Schlüssel-Dateisystems aus einem Backup (viele der einzelnen Schritte werden detailliert in Kapitel 10 erläutert):

- Booten Sie von einem alternativen Medium: entweder vom Installationsband bzw. der Installations-CD oder von einer von Ihnen angelegten Boot-Diskette bzw. einem Boot-Band (wie das geht, wird gleich noch erläutert). An diesem Punkt arbeiten Sie mit einem im Speicher liegenden (RAM-Disk) oder einem auf dem Boot-Medium vorhandenen Dateisystem.

- Generieren Sie Gerätedateien für die Festplatten, Plattenpartitionen und/oder Bandlaufwerke, auf die Sie zugreifen müssen. Möglicherweise wurden diese Gerätedateien bereits für Sie angelegt, wenn Sie ein System-Utility zur Generierung boot-fähiger Bänder bzw. Disketten verwendet haben.
- Bereiten Sie die Festplatte so weit wie nötig vor. Dazu gehören (in seltenen Fällen) die Formatierung und die Partitionierung. Stellen Sie sicher, dass Sie alle notwendigen Schritte durchführen, um eine boot-fähige Platte zu erzeugen.
- Generieren Sie bei Bedarf ein neues Dateisystem auf der entsprechenden Partition.
- Mounten Sie das System-Dateisystem (*/mnt* ist der übliche Mountpunkt).
- Wechseln Sie zum Mountpunkt. Spielen Sie die Dateien vom Sicherungsband ein. Wechseln Sie zurück ins root-Verzeichnis und deaktivieren Sie das wiederhergestellte Dateisystem mit `umount`.
- Wiederholen Sie diesen Prozess für alle weiteren Dateisysteme und starten Sie das System neu.

Es gibt noch einen weiteren Punkt, den es zu beachten gilt, wenn Sie diesen Ansatz verwenden wollen. Das auf Boot-Bändern oder -Disketten vorhandene Dateisystem ist sehr eingeschränkt, d.h., nur ein kleiner Teil der normalen Systembefehle ist verfügbar. Sie müssen also sicherstellen, dass das für die Wiederherstellung benötigte Utility nach dem Booten des alternativen Mediums auch vorhanden ist. Wenn Sie zum Beispiel eine Boot-Diskette haben, auf der sich nur `cpio` befindet, sollten Sie die Sicherung des root-Dateisystems nicht mit `tar` durchgeführt haben, andernfalls befinden Sie sich in ernsthaften Schwierigkeiten. Sie müssen darüber hinaus sicherstellen, dass alle vom gewünschten Utility benötigten Shared Libraries vorhanden sind. Überprüfen Sie das, bevor die Katastrophe eintritt.

Wir wollen uns diesen Prozess nun für jedes unserer Unix-Betriebssysteme einzeln ansehen.

AIX: mksysb und savevg

AIX stellt das Utility `mksysb` zur Verfügung, mit dessen Hilfe boot-fähige Backup-Bänder des aktuellen Systems hergestellt werden können, die sich im Falle eines Fehlers selbst wiederherstellen. Es sichert alle Dateisysteme der Volume-Gruppe root, üblicherweise */*, */usr*, */var*, */home* (wenn es nicht verschoben wurde) und */tmp*. Zusätzlich werden die Paging-Bereiche in *rootvg* gesichert. `mksysb` wird wie folgt aufgerufen:

```
# mksysb -i /dev/rmt0
```

`mksysb` ist von einer Datendatei abhängig, die verschiedene Informationen zur Systemkonfiguration enthält. Sie wird über die `mksysb`-Option `-i` aktualisiert. Verwenden Sie stattdessen die Option `-m`, wenn Sie die exakten Plattenpositionen des Dateisystems in der root-Volume-Gruppe zusammen mit deren Inhalten wiederherstellen wollen (`-m` legt fest, dass die Logical Volume Maps sowie alle anderen Konfigurationsinformationen gespeichert werden sollen).

Um die root-Volume-Gruppe wiederherzustellen, booten Sie vom mksysb-Band und wählen Sie die gewünschte Option aus dem erscheinenden Menü. Das System wird dann vom mksysb-Band wiederhergestellt.

Sie können eine ähnliche Technik verwenden, um ein System von einem mksysb-Band zu klonen, das auf einem anderen System angelegt wurde. Wenn alle Geräte identisch sind, besteht die einzige Einschränkung darin, dass Sie keinen Multiprozessor-Kernel auf einem System mit nur einer CPU installieren dürfen (und umgekehrt).

Sind die Geräte auf dem Quell- und Zielsystem unterschiedlich, wird eine leicht geänderte Technik verwendet. Zuerst booten Sie vom Installationsmedium und wählen dann die Option zur Wiederherstellung von einem mksysb-Band. In diesem Modus bindet das Betriebssystem automatisch Treiber vom Installationsmedium ein, wenn diejenigen auf dem mksysb-Band nicht zum Zielsystem passen. Beachten Sie, dass diese Methode nur dann funktioniert, wenn das Zielsystem über die Laufwerke verfügt, um das mksysb- und das Installationsmedium gleichzeitig zu verarbeiten.

Einzelne Dateien mit Hilfe eines mksysb-Bandes wiederherstellen

mksysb-Bänder können auch in nicht so dringenden Fällen als Backups des root-Volumes dienen. Es ist sehr einfach, einzelne Dateien von diesen Bändern wieder einzuspielen. Die Bänder enthalten vier verschiedene (Band-)Dateien, und die Dateien des root-Volumes befinden sich in der vierten Datei, bei der es sich um ein restore-Archiv handelt.

Um also die Datei */usr/bin/csh* und das Unterverzeichnis */etc/mf* von einem mksysb-Band wiedereinzuspielen, verwenden Sie den folgenden Befehl:

```
# restore -s 4 -x -q -f /dev/rmt0 ./bin/csh ./etc/mf
```

Die Option -s gibt an, welche Datei verwendet werden soll, und die Option -q unterdrückt dabei die anfängliche Aufforderung, die Enter-Taste zu drücken, nachdem Sie das erste Volume gemountet haben. Verwenden Sie die restore-Option -T, um den Inhalt eine Archivs einzusehen.

Sichern und Wiederherstellen von AIX-Benutzer-Volume-Gruppen

Mit dem savevg-Befehl können Sie eine vollständige Benutzer-Volume-Gruppe sichern, genau wie mksysb das für die root-Volume-Gruppe macht. Zum Beispiel sichert der folgende Befehl alle Dateien der Volume-Gruppe *chemvg* auf das Bandlaufwerk 1:

```
# savevg -i chemvg /dev/rmt1
```

Die Option -i erzeugt die Konfigurationsdatei, die zur Sicherung und Wiederherstellung der Volume-Gruppe benötigt wird. Verwenden Sie stattdessen -m, werden auch die logischen Volume Maps mitgesichert, was die Reproduktion der physikalischen Lage auf der Platte ermöglicht.

savevg kennt auch die Option -e, mit der Sie die Dateien und Verzeichnisse aus dem Sicherungssatz ausschließen können, die in der Datei */etc/exclude.vgname* aufgeführt sind.[17] Wildcards sind in diesen Ausschlusslisten nicht erlaubt.

Alle logischen Volumes und Dateisysteme sowie die Dateien innerhalb einer Volume-Gruppe können von einem savevg-Band wiederhergestellt werden. Diese Operation wird vom restvg-Utility ausgeführt. Zum Beispiel stellen die folgenden Befehle die gerade gesicherte Volume-Gruppe *chemvg* wieder her:

```
# restvg -q -f /dev/rmt1
# restvg -q -s -f /dev/rmt1 hdisk4 hdisk5
```

Der erste Befehl stellt die Volume-Gruppe auf den ursprünglichen Platten wieder her, wobei diese Operation sofort und ohne Frage nach dem ersten Band beginnt. Der zweite Befehl stellt die Struktur und den Inhalt der *chemvg*-Volume-Gruppe auf den Platten 4 und 5 wieder her. Alle logischen Volumes werden dabei auf die minimal notwendige Größe reduziert, die für die Aufnahme der Dateien notwendig ist (-s).

Das von savevg erzeugte Band enthält ein restore-Archiv, so dass einzelne Dateien einfach wieder eingespielt werden können:

```
# restore -f /dev/rmt1 -T -q
# restore -f /dev/rmt1 -x -q -d ./chem/src/h95
```

Der erste Befehl gibt den Inhalt des Archivs aus und der zweite stellt den */chem/src/h95*-Unterbaum wieder her, wobei alle notwendigen Unterverzeichnisse erzeugt werden (-d).

FreeBSD

FreeBSD stellt verschiedene Optionen zur Wiederherstellung von Systemdateien zur Verfügung, die aber alle ein vollständiges Backup des Dateisystems verlangen, über das die Wiederherstellung erfolgen soll.

Im Falle eines Platten- oder Boot-Fehlers müssen Sie von einem alternativen Medium (CD-ROM oder einer Boot-Diskette) booten. Dann wählen Sie die Fixit-Option aus dem erscheinenden Hauptmenü. An diesem Punkt können Sie wählen, ob Sie von der zweiten Installations-CD (die als Live-Dateisystem fungiert) oder einer fixit-Diskette booten wollen, oder Sie können eine eingeschränkte Shell starten. Die ersten beiden Optionen sind die nützlichsten.

Die fixit-Diskette enthält ein eingeschränktes FreeBSD-Betriebssystem, auf dem ausreichend Tools vorhanden sind, um ein Backup wiederherstellen zu können. Es unterstützt die tar- und restore-Befehle und Bandlaufwerke. Sie können eine fixit-Diskette erzeugen, indem Sie die erste Installations-CD mounten und einen Befehl wie den folgenden eingeben:

```
# dd if=/cdrom/floppies/fixit of=/dev/rfd0c bs=36b
```

Diese Diskette kann nach dem Anlegen an Ihre besonderen Bedürfnisse angepasst werden.

17 Der mksysb-Befehl erkennt -e ebenfalls und seine Ausschlussdatei ist */etc/exclude.rootvg*.

Um die Layouts der Plattenpartitionen eines FreeBSD-Systems zu sichern, verwenden Sie die Befehle `fdisk -s` und `disklabel`. Zusammen mit */etc/fstab* erlauben diese Informationen die Rekonstruktion der Plattenpartitionen und des Dateisystem-Layouts. Der `disklabel`-Befehl kann auch verwendet werden, um einen Boot-Block auf eine Ersatz-Systemfestplatte zu schreiben.

HP-UX: make_recovery

HP-UX stellt die `make_recovery`-Einrichtung zur Erzeugung boot-fähiger Recovery-Bänder als Teil des Ignite-UX-Pakets zur Verfügung (das Utility liegt in */opt/ignite/bin*). Eine gängige Methode für den Einsatz dieses Utilities sieht wie folgt aus:

```
# make_recovery -p -A -d /dev/rmt/1mn
# emacs /var/opt/ignite/recovery/arch.include
# make_recovery -r -A -d /dev/rmt/1mn -C
```

Zuerst führen wir den Befehl im Preview-Modus (`-p`) aus. Dieser Befehl schreibt keine Daten auf Band, sondern erzeugt stattdessen die Datei */var/opt/ignite/recovery/arch.include*, die aus einer Liste der aufzunehmenden Dateien besteht. Hier haben wir uns dazu entschieden, das gesamte root-Dateisystem über `-A` zu sichern. Standardmäßig wird nur der Teil des Betriebssystems gesichert, der Teil des HP-UX-Betriebssystems ist.

Sobald der Befehl abgearbeitet wurde, überprüfen wir die Logdatei */var/opt/ignite/logs/makrec.log1* auf Fehler oder Warnungen. Wenn es welche gibt, müssen wir die Aktionen durchführen, die zu ihrer Korrektur notwendig sind, und müssen dann den ersten Befehl erneut ausführen.

Sobald alle Fehler behoben wurden, kann die Datei *arch.include* editiert werden, um Elemente hinzuzufügen oder zu entfernen. Dann können Sie `make_recovery` noch einmal im so genannten Resume-Modus (`-r`) ausführen.[18] Die Option `-C` weist den Befehl an, die Daten der letzten `make_recovery`-Prozedur zu aktualisieren.

Dieser Prozess muss nach jeder bedeutenden Systemänderung wiederholt werden. Mit dem Befehl `check_recovery` können Sie ermitteln, ob make_recovery ausgeführt werden muss.

Obwohl diese Bänder nicht als Ersatz für normale Backups gedacht sind, ist es möglich, einzelne Dateien daraus wiederherzustellen. Zu diesem Zweck müssen Sie das Band von Hand an der zweiten Datei positionieren und die gewünschten Elemente dann mit `tar` extrahieren:

```
# cd /
# mt -t /dev/rmt/1mn fsf 1
# tar xvf /dev/rmt/1m relative(r)-pfadname(n)
```

Die Dateiliste muss mit relativen Pfadnamen angegeben werden (z.B. *etc/hosts*, nicht */etc/hosts*).

18 In manchen Fällen sind zusätzliche Dinge zu berücksichtigen, wenn einige Systemdateien außerhalb der root-Volume-Gruppe liegen. Details finden Sie in der Manpage.

Die neuesten Versionen des HP Ignite-UX-Pakets stellen auch `make_tape_recovery` (erzeugt Bänder mit Recovery-Images des Clientsystems und des Ignite-UX-Servers) und `make_net_recovery` (schreibt ein Recovery-Image über das Netzwerk auf das Diskettenlaufwerk des Ignite-UX-Servers) zur Verfügung. Details finden Sie in der Dokumentation.

Linux

Auf Linux-Systemen können Sie eine Boot-Diskette des aktuellen Kernels mit folgendem Befehl erzeugen:

```
# dd if=/boot/Datei of=/dev/fd0
```

Das einfache Kopieren des komprimierten Kernels auf die Diskette ist alles, was notwendig ist, weil der Linux-Kernel so strukturiert ist, dass er das Image einer boot-fähigen Diskette aufweist (und entweder über den DOS-Boot-Loader oder von `lilo` geladen werden kann).

Diese Prozedur ermöglicht Ihnen das Booten Ihres Systems, falls es ein Problem mit dem Booten von der Festplatte geben sollte. Ist Ihre System-Festplatte allerdings beschädigt und das darauf enthaltene root-Dateisystem nicht zugänglich, benötigen Sie ein echtes Recovery-System, um die Dinge wiederherzustellen. In solchen Fällen können Sie von einer *Notfalldiskette* (rescue disk) booten. Um eine solche Diskette zu erzeugen, müssen Sie die Installations-CD mounten und den folgenden Befehl eingeben:

```
# dd if=/cdrom/disks/rescue of=/dev/fd0 bs=18k
```

Diese Notfalldiskette enthält die Werkzeuge, die zur Wiederherstellung eines Backups notwendig sind, inklusive eines Bandgerätes und des `tar`-Befehls.

Um die Informationen zur Partitionierung der Platte zu speichern, verwenden Sie den Befehl `fdisk -l`. Zusammen mit */etc/fstab* ermöglichen diese Informationen die Rekonstruktion der Plattenpartitionen und des Dateisystem-Layouts. Sie können `lilo` verwenden, um einen Boot-Block auf der Ersatzplatte anzulegen. Beachten Sie, dass dessen Option `-r` sehr nützlich ist, wenn die neue Partition an irgendeinem anderen Punkt (z.B. */mnt*) im Notfall-Dateisystem gemountet wurde.

Die jüngsten Versionen von Red Hat Linux bieten auch beim Booten von der Installations-CD eine Rescue-Option für den Notfall an.

Solaris

Solaris bietet nur wenige Tools für ein System-Backup und -Recovery an. Sie müssen vollständige Backups des root-Dateisystems anlegen. Sie können dann von einem alternativen Medium booten, um ein funktionierendes Minimalsystem anzulegen, und müssen Ihr Backup dann wieder einspielen.

Der `prtvtoc`-Befehl zusammen mit */etc/checklist* liefert die Informationen, die notwendig sind, um die Plattenpartitionierung und das Dateisystem-Layout wiederherzustellen. Sie können den Befehl `installboot` verwenden, um einen Boot-Block auf die Systemplatte zu schreiben. Beachten Sie, dass Boot-Images innerhalb des installierten Dateisystems unter */usr/plattform/modell/lib/fs/ufs/bootblk* abgelegt sind, wobei *modell* ein String ist, der Ihrer jeweiligen Sun-Hardware entspricht (z.B. *SUNW-Sun-Blade-100*).

Tru64: btcreate

Tru64 stellt den Befehl `btcreate` zur Generierung eines boot-fähigen Backup-Bandes des Betriebssystems zur Verfügung. Das Band besteht aus einem boot-fähigen Mini-Betriebssystem und einem vollständigen Backup der Systemdateien.

Die Ausführung von `btcreate` ist sehr einfach, da Sie nach allen notwendigen Informationen gefragt werden. Die (empfohlenen) Standardantworten sind nahezu immer korrekt. Ein Restore über ein `btcreate`-Band spielt nicht nur alle Systemdateien wieder ein, sondern regeneriert auch die Konfiguration der logischen Volumes des Originalsystems.

Auf Tru64-Systemen können Sie den Befehl `disklabel -r` verwenden, um Informationen zur Plattenpartitionierung zu speichern und bei Bedarf wiederherzustellen.