

8 *So verwalten Sie Ihre Systemsicherung*

Der Datensicherung kommt eine wichtige Bedeutung zu, über die aber häufig hinweggesehen wird. Gelegentlich geschieht etwas Unerwartetes & eine Datei wird versehentlich gelöscht, oder der Strom fällt aus, während Sie an einem Dokument arbeiten. Eine häufige, aber schlechte Gewohnheit besteht darin, die Backups hinauszuzögern, da die Prozedur zu umständlich oder zeitaufwendig erscheint. Viele Anwender & aus Erfahrung klug & können Ihnen sagen, daß eine regelmäßige Datensicherung bei der Systemverwaltung unerlässlich ist. In diesem Kapitel finden Sie einige Vorschläge, um diese Aufgabe zu vereinfachen.

Backup-Strategien

Es ist wichtig, Backups systematisch und zeitlich zu planen. Wie häufig Sie eine Sicherheitskopie anlegen und diese anschließend an einem sicheren Ort aufbewahren, hängt von verschiedenen Faktoren ab & etwa wie oft Sie Ihre Dateien verändern, wieviel Zeit ein Backup erfordert, oder wie viele Daten Sie zu verlieren bereit sind. Ein vollständiges Backup Ihres Systems ist zwar am gründlichsten, erfordert aber soviel Zeit und Speicherplatz, daß es als tägliche Routine etwas unpraktisch sein könnte. Da zahlreiche Dateien ohnehin selten verändert werden, können komplette Backups seltener (etwa einmal im Monat) durchgeführt werden. Dagegen sollten Sie häufig Sicherheitskopien von den Dateien anlegen, die Sie regelmäßig ändern.

Sie können beispielsweise eine mehrstufige Backup-Strategie verwenden. Dabei werden jeweils verschiedene Dateien kopiert:

- Ein *vollständiges Backup* kopiert sämtliche Daten Ihres Systems. Sie können jedoch bestimmte Verzeichnisse oder Dateisysteme auslassen, deren Inhalt Sie nicht sichern möchten (z. B. **/tmp**).
- Bei einem *Zuwachs-Backup* werden nur Dateien kopiert, die seit dem letzten Backup verändert wurden. Manche Programme (z. B. **dump**) enthalten eine integrierte Funktion für Zuwachs-Backups.
- Ein *teilweises Backup* kopiert lediglich die angegebenen Dateien oder Verzeichnisse.

Falls Sie an einem Standalone-Computer arbeiten, kann Ihre Backup-Strategie sehr einfach sein. Kopieren Sie jeden Abend Ihre Arbeitsergebnisse auf eine Diskette. Zusätzlich sollten Sie eventuell regelmäßig (wöchentlich oder monatlich) ein vollständiges Backup durchführen.

Für Netzwerkverwalter wird die Backup-Strategie jedoch wesentlich komplexer. Die sicherste Backup-Strategie für ein Netzwerk besteht darin, routinemäßig mehrstufige Backups durchzuführen, die sich gegenseitig überlappen. Eine angemessene Strategie könnte folgendermaßen aussehen:

- Einmal im Monat führen Sie ein vollständiges Backup durch. Es ist ratsam, die Datenträger mit diesen Backups an einem anderen, sicheren Ort aufzubewahren.
- Einmal in der Woche führen Sie ein Zuwachs-Backup der Dateien durch, die seit dem letzten monatlichen Backup verändert wurden. Die dafür benutzten Datenträger können Sie allmonatlich wiederverwenden.
- Jeden Tag führen Sie ein Zuwachs-Backup aller Dateien durch, die seit dem letzten wöchentlichen Backup verändert wurden. Gleichzeitig erstellen Sie separat ein vollständiges Backup der Verwaltungsdaten in **/private/etc** und **/private/adm** (hier sind die NetInfo-Datenbanken sowie administrative Logbücher und Skripte enthalten). Die für diese Backups benutzten Datenträger können Sie jede Woche wiederverwenden.

Warnung: Benutzen Sie für zwei aufeinanderfolgende Sicherheitskopien nicht die gleichen Bänder oder Disketten. Auf diese Weise vermeiden Sie, daß das vorherige Backup zerstört wird, falls Ihr heutiges Backup fehlschlägt. Aus dem gleichen Grund sollten Sie sich immer erst vergewissern, daß Ihr neuestes Backup lesbar ist, bevor Sie das vorherige löschen.

Um Ihre Verwaltungsarbeit etwas einzuschränken, können Sie mit **cron** regelmäßig automatische Backups vornehmen. Fügen Sie Ihre Backup-Befehle in die **cron**-Skripte **daily**, **weekly** und **monthly** ein. Die Skripte finden Sie in **/usr/adm**. Mit **cron** wird das Backup **/usr/adm/daily** täglich um 2.00 Uhr ausgeführt, **/usr/adm/weekly** jeden Samstag um 3.30 Uhr und **/usr/adm/monthly** am Monatsersten um 5.30 Uhr.

Mit **cron** können Ihre Backups unbeaufsichtigt ablaufen. Legen Sie den Backup- Datenträger jeden Abend in das für diesen Zweck vorgesehene Laufwerk ein. Am nächsten Morgen entfernen und beschriften Sie den Datenträger. Bewahren Sie ihn anschließend an einem sicheren Ort auf. Bei einer gut strukturierten Planung brauchen Sie nicht mehrere Datenträger zu verwenden.

Wovon Sie Sicherungskopien anlegen sollten

Erstellen Sie Backups aller Daten, deren Verlust sehr schwerwiegend wäre. Für manche Dateien müssen jedoch häufige Backups durchgeführt werden, wenn sie beispielsweise ständig bearbeitet werden oder äußerst wichtige Informationen enthalten. Legen Sie besonderen Wert auf die nachfolgend aufgeführten Dateien:

- Dokumente, die Sie bearbeitet haben – Es ist ratsam, von diesen Dateien jeden Abend Sicherungskopien anzulegen. Auf diese Weise können Sie nicht mehr als einen Tag Arbeit verlieren. Beachten Sie dabei, daß auch ein Ausdruck eine Art Sicherungskopie ist. Ein Ausdruck kann zwar nicht so einfach wiederhergestellt werden wie eine elektronische Datei. Trotzdem vermeiden Sie damit den totalen Verlust Ihrer Arbeit.
- Heimverzeichnisse – Wenn Sie lediglich Ihre eigenen Arbeiten sichern, reicht meistens ein wöchentliches Backup des Heimverzeichnisses aus – vorausgesetzt, Sie sichern besonders wichtige Dateien täglich. Als Netzwerkverwalter sollten Sie die Heimverzeichnisse der Benutzer jedoch häufiger sichern (am besten täglich).
- **/private/etc** und **/private/adm** – Backups dieser Verzeichnisse schützen die wichtigsten Elemente der Systemdatenbanken. Diese Verzeichnisse enthalten beispielsweise die Verwaltungsinformationen über Benutzer-Accounts, einschließlich deren verschlüsselte Paßwörter. Für diese Verzeichnisse sollten separate und komplette Backups durchgeführt werden, um die interne Konsistenz der NetInfo-Datenbank zu

gewährleisten.

Eine NetInfo-Datenbank besteht aus separaten Dateien. Sie müssen darauf achten, daß diese Dateien zusammen und gleichzeitig gesichert werden. Das Programm **tar** ist dazu besonders geeignet, weil Sie damit alle Dateien auf einmal kopieren können. Bei Zuwachs-Backups ð wie sie etwa das Programm **dump** erstellt ð werden die Dateien eventuell zu verschiedenen Zeitpunkten kopiert. Dadurch könnten sich Inkonsistenzen ergeben. Netzwerkverwalter sollten tgliche Backups der Verzeichnisse **/private/etc** und **/private/adm** durchfhren.

- Verzeichnisse, die hufig verndert werdenAlle Dateien, die hufig verndert werden, sollten ebenso hufig gesichert werden. Die folgenden Verzeichnisse gehren zu dieser Kategorie: **/LocalLibrary**, **/LocalApps** und **/LocalAdmin** (falls vorhanden), **/usr/local** und **/usr/spool/mail**. Sichern Sie auch alle anderen Dateien oder Verzeichnisse, die Sie ndern, wie etwa Verzeichnisse fr gemeinsame Projekte.
- Wichtige, spezifische Dateien Ihrer AnlageLegen Sie Backups aller weiteren Dateien an, die Sie nach einer Katastrophe (etwa nach einem Brand oder Hochwasser) wiederherstellen mten. Bewahren Sie diese Kopien an einem sicheren Ort auerhalb der Anlage auf.

Da die System-Software in der Regel nicht oft verndert wird, jedoch sehr viel Speicherplatz bentigt, kann sie bei der regulren Backup-Planung ausgelassen werden. Sie sollten sich jedoch vergewissern, da eine einwandfreie Kopie der System-Software jederzeit verfgbar ist. Auerdem sollten Sie immer die ntigen Mittel verfgbar haben, um die System-Software von der Archivkopie wiederherzustellen.

Methoden der Datensicherung

Backup-Techniken knnen sowohl sehr einfach als auch uerst komplex sein. Sie knnen z. B. ein kleines Backup durchfhren, indem Sie Ihre Datei oder Ihr Verzeichnis einfach auf einen anderen Computer im Netz oder auf ein anderes Speichergert kopieren (wie beispielsweise eine optische Platte, ein zweites Festplattenlaufwerk oder eine Diskette).

Für kompliziertere Backups stehen verschiedene Programme zur Verfügung, die alle Vor- und Nachteile haben. Im folgenden Abschnitt werden sechs Programme vorgestellt: **dump** und **restore** (gemeinsam verwendet), **rdump** und **rrestore** (gemeinsam verwendet), **tar** und **cpio**.

Die folgenden Hinweise sollten Sie bei Backup-Prozeduren beachten:

- Melden Sie sich als **root** an, als ein Mitglied der Gruppe **operator** (die beste Wahl) oder als Eigentümer der Dateien, von denen Sie ein Backup anlegen oder die Sie wiederherstellen wollen. Um Daten wiederherzustellen, benötigen Sie eine Schreibberechtigung für das Verzeichnis, in dem sie wiederhergestellt werden sollen. Sie sollten die Benutzer-Accounts der Benutzer, die Backups durchführen, der Gruppe **operator** hinzufügen. Diese Benutzer können dann Sicherungskopien anlegen, ohne sich speziell als **root** anzumelden. Beachten Sie, daß **root** für entfernte NFS-Verzeichnisse eventuell keine Zugriffsberechtigung hat.
- Sichern Sie die wichtigsten Daten zuerst. Falls während des Backups etwas fehlschlägt oder Ihr Speichermedium nicht über ausreichend Platz verfügt, sind die wichtigsten Daten auf jeden Fall gesichert. So sollten Sie Benutzerdaten normalerweise vor Systemdaten sichern, da sie häufiger verändert werden. Die Systemdateien in **/etc** werden jedoch häufig verändert und sind außerordentlich wichtig. Sie sollten diese Dateien deshalb stets zuerst sichern.
- Führen Sie Ihre Datensicherung möglichst im Einzelbenutzer-Modus durch oder zu einem Zeitpunkt, an dem niemand das System benutzt. (Es ist nicht ratsam, Dateien zu verändern, während ein Backup davon erstellt wird.) Normalerweise werden Backups nachts erstellt.
- Beschriften Sie Ihre Datenträger sorgfältig. Das Etikett sollte Angaben über Datum, Uhrzeit und Art des Backups sowie eine Inhaltsangabe enthalten. Auf diese Art sparen Sie Zeit, wenn Sie Daten wiederherstellen müssen. Sie sollten außerdem in einem separaten Logbuch alle Vorgänge schriftlich festhalten. Denken Sie daran, daß Sie diese Informationen eventuell gerade dann benötigen, wenn Ihr System abgestürzt ist. Speichern Sie sie deshalb auf keinen Fall online.
- Erstellen Sie eine Datei mit dem Inhaltsverzeichnis des Backups. Dies sollte stets die erste Datei auf einem Band sein.
- Aktivieren Sie den Schreibschutz des Datenträgers. So verhindern Sie, daß er versehentlich überschrieben

- wird. Bevor Sie den Datenträger wiederverwenden, müssen Sie den Schreibschutz deaktivieren.
- Bewahren Sie Ihre Datenträger an einem sicheren Ort auf, zu dem nur befugte Personen Zugang haben.

Die Programme **^dump^** und **^restore^**

Mit dem Programm **dump** können Sie Zuwachs-Backups Ihrer Dateien vornehmen. Mit dieser Methode werden nur die Dateien kopiert, die seit dem letzten Backup verändert wurden. Da ein großer Teil der Daten im System nur selten verändert wird, können Sie auf diese Weise die Zahl der zu sichernden Daten stark reduzieren. Mit dem Programm **restore** können Sie Dateien wiederherstellen, von denen Sie mit **dump** eine Sicherungskopie erstellt haben.

Wichtig: Sie sollten die Programme **dump** und **restore** von einem Account aus starten, das zur Gruppe **operator** gehört, jedoch *nicht* vom **^root^**-Account aus. Diese Vorsichtsmaßnahme stellt sicher, daß Sie die Platte nicht versehentlich überschreiben, falls Ihnen bei der Ausführung von **dump** oder **restore** ein Tippfehler unterläuft.

Die Vorteile von **^dump^** und **^restore^**:

- Eine einfache Methode, große Datenmengen zu speichern und wiederherzustellen.
- Es können interaktive Backups und Wiederherstellungen ausgeführt und kontrolliert werden.
- Es können Zuwachs-Backups ausgeführt werden.
- Die Programme sind schnell.

Die Nachteile von **^dump^** und **^restore^**:

- Beide können nur für vollständige Dateisysteme verwendet werden. Sie können die Programme nicht verwenden, um nur einzelne Dateien und Verzeichnisse zu sichern.

So verwenden Sie **^dump^**

Das Programm **dump** registriert, wie umfassend ein Backup sein sollte. Dazu ordnet das Programm jeder Backup-Stufe eine Ziffer zwischen 0 und 9 zu. Ein **dump** der Stufe 0 kopiert das gesamte Dateisystem. Mit höheren **dump**-Stufen werden nur Dateien gesichert, die seit dem letzten **dump** mit einer niedrigeren Stufe verändert wurden. Mit Stufe 5 werden beispielsweise alle Dateien kopiert, die seit dem letzten **dump** der Stufe 4 oder niedriger verändert wurden. Falls Sie **dump** mit der Option **u** ausführen, zeichnet das Programm Datum, Uhrzeit und Stufe auf, mit der das Dateisystem gesichert wurde und zwar in der Datei **/etc/dumpdates**.

Eine einfache Backup-Strategie besteht darin, mit einem **dump** der Stufe 0 anzufangen. Erstellen Sie dann täglich Zuwachs-Backups.

Warnung: Bevor Sie **dump** mit Stufe 0 durchführen, sollten Sie **fsck** verwenden. Dadurch stellen Sie sicher, daß Sie ein sauberes Dateisystem kopieren. Wenn Sie diesen Schritt übergehen, könnten Sie das Dateisystem beschädigen. Verwenden Sie **fsck** nur für Dateisysteme, für die kein *mount* durchgeführt wurde. Außerdem sollte für das Dateisystem der *unmount* durchgeführt werden, wenn Sie **dump** der Stufe 0 durchführen. Dies stellt sicher, daß Sie ein vollständiges und konsistentes Backup erhalten. Falls Sie ein Backup des Root-Dateisystems erstellen, sollten Sie dabei den Einzelbenutzer-Modus wählen. Weitere Informationen über **fsck** finden Sie in Kapitel 9, *Hochfahren und Herunterfahren des Systems*, sowie im UNIX-Handbuch unter **fsck**.

Beispiele für die Verwendung von *dump*

1. Um ein komplettes Backup der Stufe 0 durchzuführen, geben Sie einen Befehl ähnlich dem folgenden ein:

```
dump 0u /dev/rsd0a
```

Mit diesem Befehl legen Sie eine vollständige Sicherheitskopie von **/dev/rsd0a** an. Die Ausgabe erfolgt auf das voreingestellte Backup-Gerät (**/dev/rxt0**, ein externes SCSI-Bandlaufwerk).

Warnung: Führen Sie **dump** der Stufe 0 durch, dürfen Sie *auf keinen Fall* Ihr letztes Backup überschreiben. Bewahren Sie das letzte Backup auf und schreiben Sie das neue auf ein anderes Band. So wird Ihr einziges Backup der Stufe 0 vor Zerstörung geschützt (falls beispielsweise während des Backups der Strom ausfällt).

2. Mit einem Befehl ähnlich dem folgenden legen Sie tägliche, wöchentliche oder monatliche Backups an (ersetzen Sie 9 durch die entsprechende Backup-Stufe und *dateisystem* durch den entsprechenden

Gerätenamen):

```
dump 9u dateisystem
```

Denken Sie daran, daß Sie Ihre Backup-Befehle in ein **cron**-Skript einfügen können, damit sie automatisch ausgeführt werden. Beschriften Sie Ihre **dump**-Befehle sehr sorgfältig.

Hinweis: Es spielt keine Rolle, welche Nummern Sie verwenden, um die einzelnen Backup-Stufen zu kennzeichnen. Sie müssen jedoch immer drei verschiedene Nummern verwenden und diese konsequent beibehalten. Sie können beispielsweise Nummer 1 für das monatliche, Nummer 2 für das wöchentliche und Nummer 9 für das tägliche Backup verwenden oder 1 für monatlich, 4 für wöchentlich und 7 für täglich.

3. Wenn Sie eine andere Ausgabedatei oder ein anderes Ausgabegerät angeben möchten, verwenden Sie die Option **f**:

```
dump 9uf /Dumps/SD1a-9.dump /dev/rsd1a
```

In diesem Beispiel erfolgt das Backup von **/dev/rsd1a** mit Stufe 9 und wird in die Datei **/Dumps/SD1a-9.dump** geschrieben. Diese Datei könnte sich auf einer Wechselplatte befinden (vorausgesetzt, für die Wechselplatte wurde der `mount` in **/Dumps** durchgeführt). Die Namensweiterung **.dump** identifiziert die Datei als **dump**-Backup.

4. Um Platz auf dem Backup-Datenträger zu sparen, können Sie die Ausgabe komprimieren:

```
dump 9uf - /dev/rsd0b | compress > /Dumps/SD0b-9.dump.Z
```

Wenn Sie `-` als Ausgabedatei angeben, wird das Backup an die Standardausgabe (**stdout**) gesendet. Die Ausgabe von **dump** wird dann als Eingabe für den Befehl **compress** verwendet. Schließlich wird die Ausgabe von **compress** an die Datei **/Dumps/SD0b-9.dump.Z** umgeleitet. (Die Namensweiterung **.Z** wird standardmäßig verwendet, um anzuzeigen, daß die Datei komprimiert wurde.) Mit dem Befehl **uncompress** oder **zcat** wird die ursprüngliche **dump**-Datei vollständig wiederhergestellt.

5. Da die System-Software normalerweise nie verändert wird, können Sie sie beim regulären, vollständigen Backup auslassen. Dadurch verringern Sie den Speicherplatz, der für ein vollständiges Backup des Root-Dateisystems erforderlich ist. Vergewissern Sie sich aber, daß Sie über die nötigen Mittel verfügen, um die System-Software bei Bedarf neu zu installieren. Geben Sie anschließend folgenden Befehl ein:

```
dump 0uf /dev/null /dev/rsd0a
```

Warnung: Falls Sie diese Methode wählen, müssen Sie dieses *„falsche“* **dump** der Stufe 0 durchführen, *bevor* Sie irgendwelche Veränderungen an einem neuen System vornehmen (oder sofort nach Installation der System-Software, falls Sie eine andere Platte als Startplatte verwenden).

Dieser Befehl aktualisiert **/etc/dumpdates** und zeigt an, daß die **dump**-Stufe 0 verwendet wurde, ohne tatsächlich eine Sicherungskopie anzulegen. (Die Ausgabe wird an **/dev/null** gesendet, das *Null-Gerät*, oder auch *„bit bucket“* (Bit-Eimer) genannt wird. Alle dorthin gesendeten Ausgaben werden verworfen.) Für vollständige Backups Ihres Root-Dateisystems verwenden Sie von jetzt an die **dump**-Stufe 1. Der Konsistenz halber können Sie vollständige Backups Ihrer anderen Dateisysteme dann ebenfalls mit **dump**-Stufe 1 durchführen & vorausgesetzt, Sie erstellen von diesen Daten nie ein Backup der **dump**-Stufe 0.

Im UNIX-Handbuch unter **dump** finden Sie eine vollständige Liste sowie Beschreibungen aller Argumente, die Sie mit **dump** verwenden können.

So verwenden Sie *„restore“*

Möchten Sie Daten wiederherstellen, die Sie mit **dump** gesichert haben, verwenden Sie **restore**. Das Programm **restore** kopiert Daten von der **dump**-Datei in Ihr aktuelles Verzeichnis. In der Voreinstellung verwendet **restore** genauso wie **dump** ein externes Bandlaufwerk (**/dev/rxt0**). Sie können dies jedoch mit der Option **f** ändern.

Warnung: Seien Sie mit **restore** besonders vorsichtig, damit Sie vorhandene Daten nicht versehentlich überschreiben. Dateien, die Sie mit **restore** wiederherstellen, werden in das aktuelle Arbeitsverzeichnis geschrieben. Bereiten Sie für die wiederhergestellten Daten ein leeres Verzeichnis vor. Sie sollten Daten *nie* in einem Verzeichnis wiederherstellen, das bereits andere Dateien enthält, es sei denn, Sie tun dies bewußt. Die Option **i** kann zusammen mit **restore** besonders nützlich sein. Sie wird weiter hinten in diesem Abschnitt beschrieben.

Sie müssen **restore** immer zusammen mit einem Schlüsselargument verwenden. Dieser Schlüssel bestimmt die Wirkung von **restore** ganz genau. Nachfolgend finden Sie einige der wichtigsten Schlüsselargumente:

- **r** Liest die gesamte **dump**-Datei in das aktuelle Verzeichnis ein. Dieser Schlüssel sollte *ausschließlich* verwendet werden, um eine komplette **dump**-Datei in einem leeren Dateisystem wiederherzustellen & oder

zur Wiederherstellung der **dump**-Datei eines Zuwachs-Backups, nachdem ein vollständiges Backup der Stufe 0 wiederhergestellt wurde.

- **x** *dateiname* Extrahiert die angegebene Datei oder das angegebene Verzeichnis aus einer **dump**-Datei. Beachten Sie, daß **restore** sämtliche Verzeichnisse erstellen muß, die zur Datei führen. Erst dann kann die Datei selbst wiederhergestellt werden. Dies müssen Sie berücksichtigen, wenn Sie das Verzeichnis bestimmen, in dem Sie die Datei wiederherstellen möchten. Dieses Verzeichnis muß eine umfangreiche Verzeichnishierarchie aufnehmen können und nicht nur eine einzelne Datei. Vorzugsweise sollten die Dateien in einem vorläufigen Verzeichnis wiederhergestellt werden. Kopieren oder verschieben Sie die wiederhergestellten Dateien dann in das endgültige Verzeichnis.
- **t** [*dateiname*] Zeigt ein Inhaltsverzeichnis für die **dump**-Datei an. Mit dem Argument *dateiname* wird das Inhaltsverzeichnis der angegebenen Datei angezeigt. Ohne dieses Argument zeigt **restore** den Inhalt des gesamten Backups an.
- **i** Führt eine interaktive Wiederherstellung durch und ist besonders für begrenzte, selektive Wiederherstellungen nützlich. Sie können mit diesem Befehl auch den Inhalt der **dump**-Datei einsehen. Bei einer interaktiven Wiederherstellung geben Sie mehrere Befehle über eine Schnittstelle ein, die einer Shell ähnelt.

Beispiele für die Verwendung von **restore**

1. Nachfolgend finden Sie ein kommentiertes Beispiel einer interaktiven **restore**-Sitzung:

```
mkdir /new
cd /new
/etc/restore if /dev/rxt0
restore > ls
.:
./          LocalAdmin/
../         LocalApps/
.NeXT/      LocalDeveloper/
.cshrc      LocalLibrary/
.hidden     Net/
```

Führt beim **restore**-Prompt einen Befehl
(ls) aus.

```
files/      me/
fun.eps     mnta/
bin/        odmach
cores/      private/
dev/        savem
```

.login	NextAdmin/	etc/	sdmach
.profile	NextApps/	lib/	tmp/
.rhosts	NextDeveloper/	lost+found/	usr/
.alias	NextLibrary/	mach@	

```
restore > cd /files
restore > ls
./files:
bin/
```

Wechselt in ein bestimmtes Verzeichnis des Bands über und listet den Inhalt auf.

```
restore > cd /bin
restore > ls
./files/bin:
sh
```

```
restore > add sh
```

Fügt die Datei **sh** der Liste der wiederherzustellenden Dateien hinzu.

```
restore > extract
```

Alle Dateien der Liste werden vom **dump**-Band wiederhergestellt.

```
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 towards the first.
Specify next volume #: 1
```

Gibt das Band an, von dem das Programm `restore` beginnen soll.

```
set owner/mode for '.'? [yn] y
```

Gibt den Benutzer, der `restore` durchführt, als Eigentümer an und stellt die voreingestellten Berechtigungen wieder her.

```
restore > quit
```

Verläßt das Programm.

- Im folgenden Beispiel wird eine vollständige Wiederherstellung des Backups auf einer neu formatierten Platte durchgeführt (dies ist nützlich, wenn Sie einen Upgrade auf eine neue Platte durchführen):

```
disk -i /dev/rsd1a
```

Formatiert die neue Platte.

mount /dev/sd1a /mnt

cd /mnt

restore rf /dev/rst0

rm restoresymtab

Führt den **mount** für ein Dateisystem auf der Platte durch.

Wechselt zu dem Verzeichnis, für das der **mount** durchgeführt wurde.

Stellt den gesamten Inhalt der dump-Datei wieder her.

Löscht eine von **restore** zurückgelassene vorläufige Datei.

3. Der folgende Befehl führt eine interaktive Wiederherstellung eines komprimierten Backups aus:

zcat /Dumps/SD0a-9.dump.Z | **restore if** -

Weitere Informationen über **restore** und seine zahlreichen Argumente finden Sie im UNIX-Handbuch unter **restore**.

Die Programme **rdump** und **rrestore**

Die Programme **rdump** und **rrestore** funktionieren mehr oder weniger in der gleichen Weise wie **dump** und **restore**. Sie werden jedoch verwendet, um ein Dateisystem über das Netzwerk auf einem entfernten Gerät zu sichern (etwa auf einem gemeinsam genutzten Bandlaufwerk).

Vorteile von **rdump** und **rrestore**:

- Die Programme können sich praktischer als **dump** erweisen, da Sie mit einem einzigen Bandlaufwerk Backups und Wiederherstellungen für alle Computer im Netzwerk durchführen können.
- Die Programme haben ansonsten dieselben Vorteile wie **dump** und **restore**.

Nachteile von **rdump** und **rrestore**:

- Die Programme haben dieselben Nachteile wie **dump** und **restore**.

Hinweis: Eventuell möchten Sie mit **rdump** ein Backup einer Platte erstellen, die an ein entferntes System angeschlossen ist. In diesem Fall müssen Sie die Berechtigung haben, den Befehl **rsh** als **root** auf dem

Computer auszuführen, an dem das Zielausgabegerät angeschlossen ist. Diese Berechtigung legen Sie fest, indem Sie für Ihren entfernten Computer einen Eintrag in der Datei **/etc/hosts** erstellen. Wenn Sie z. B. ein Backup einer Platte anlegen, die an den Computer **bilbo** angeschlossen ist, und die Ausgabedatei an eine Platte senden, die an den Computer **gandalf** angeschlossen ist, müssen Sie für **bilbo** einen Eintrag in der Datei **/etc/hosts** von **gandalf** erstellen. Im UNIX-Handbuch unter **.hosts** finden Sie weitere Informationen zu diesem Thema.

Sie verwenden **rdump** in derselben Weise wie **dump**, nur müssen Sie jedesmal das entfernte Gerät angeben. Dazu verwenden Sie die Option **f** gefolgt von einem Argument in der Form von *host:gerät*.

Möchten Sie Daten, die mit **rdump** kopiert wurden, über ein Netzwerk wiederherstellen, verwenden Sie **rrestore**. Dieses Programm funktioniert fast genauso wie **restore**, nur müssen Sie immer das entfernte Gerät angeben, von dem Sie die Daten wiederherstellen. Dazu verwenden Sie den Schlüssel **f** gefolgt von einem Argument in der Form von *host:gerät*. So stellen Sie Daten vom Laufwerk *gerät* auf dem Computer *host* wieder her. (Vergewissern Sie sich auch hier, daß die Datei **/etc/hosts** auf dem entfernten Computer einen entsprechenden Eintrag enthält.)

Wie **restore** stellt auch **rrestore** Daten immer im aktuellen Verzeichnis wieder her. Folglich müssen Sie sich bei dem Computer anmelden, auf dem Sie die Daten wiederherstellen möchten, und in das entsprechende Verzeichnis überwechseln. Erst dann können Sie **rrestore** ausführen. Falls Sie nicht an dem Computer sitzen, der die Daten aufnehmen wird, können Sie sich mit den Befehlen **rlogin** oder **telnet** anmelden. Weitere Einzelheiten hierzu finden Sie im UNIX-Handbuch unter **rlogin** und **telnet**.

Beispiele für die Verwendung von **rdump** und **rrestore**

1. Das folgende Beispiel schreibt ein Zuwachs-Backup von **/dev/rsd2a** auf das Bandlaufwerk **/dev/nrst0** des Hosts **tserver**:

```
rdump 9uf tserver:/dev/nrst0 /dev/rsd2a
```

2. Dieses Beispiel stellt die Daten vom Band in Laufwerk **/dev/rst0** des Hosts **tserver** interaktiv wieder her:

```
rrestore if tserver:/dev/rst0
```

Weitere Informationen über **rdump** und **rrestore** finden Sie im UNIX-Handbuch unter **rdump**, **dump**, **rrestore** und **restore**.

Das Programm **tar**

Das Programm **tar** speichert mehrere Dateien und Verzeichnisse in einer einzigen großen Archivdatei. Seine Funktionen werden von einem Schlüsselargument gesteuert. Sie können auch Datei- oder Verzeichnisnamen angeben und damit festlegen, welche Dateien kopiert oder wiederhergestellt werden.

Vorteile von **tar**:

- Ermöglicht es Ihnen, Kopien von einzelnen Dateien und Verzeichnissen zu machen.
- Das Programm ist schnell.
- **tar** ist auch mit anderen Dateisystemen außer UNIX kompatibel. (Beispielsweise könnten Sie mit **tar** Daten von einer DOS-Diskette kopieren und sie dann auf einer anderen DOS-Diskette wiederherstellen.)

Nachteile von **tar**:

- Kann komplexere Operationen, wie etwa Zuwachs-Backups, nicht durchführen.
- Beschränkt die Pfadnamen von Dateien auf 100 Zeichen.

Warnung: Da **tar** nicht mit mehreren Datenträgern arbeiten kann, sollten Sie es nicht verwenden, wenn das zu sichernde Material nicht auf einen Datenträger paßt. Das Programm **tar** kopiert auch keine Dateien, deren Namen länger als 100 Zeichen sind. Dies hat besondere Bedeutung für NeXT-Computer, da manche Dateien sehr lange Namen haben können.

Der Befehl **tar** kann mit verschiedenen Schlüsseln gesteuert werden. Nachfolgend werden die häufigsten aufgeführt:

- **c** *dateiname* **Ö** Erstellt eine neue **tar**-Datei und beginnt den Schreibvorgang von ihrem Anfang an. (Standardmäßig liest und schreibt **tar** auf einem externen Bandlaufwerk **/dev/rxt0**.) Wenn Sie ein Verzeichnis angeben, kopiert **tar** die gesamte Verzeichnisstruktur, nicht nur die oberste Ebene.

- **x** *dateiname* ⚭ Die angegebene Datei wird aus der **tar**-Datei extrahiert. Wenn Sie das Argument *dateiname* auslassen, stellt **tar** den gesamten Inhalt der **tar**-Datei wieder her. Seien Sie vorsichtig, damit vorhandene Daten auf diese Art nicht überschrieben werden.
- **p** ⚭ Die Berechtigungen der extrahierten Dateien werden beibehalten.
- **t** ⚭ Listet den Inhalt der **tar**-Datei auf, ohne die Dateien tatsächlich zu extrahieren.
- **h** ⚭ Mit diesem Schlüssel folgt **tar** symbolischen Verweisen (es kopiert die Datei, auf die verwiesen wird ⚭ nicht nur die Verweisdatei).
- **v** ⚭ Mit diesem Schlüssel zeigt **tar** den Namen jeder Datei an, die kopiert wird. Kombiniert mit der Option **t** erhalten Sie noch mehr Informationen über die Dateien.

Wenn Sie mit **tar c** eine Archivkopie erstellen, geben Sie die Verzeichnisse oder Dateien, die kopiert werden sollen, entweder mit einem *relativen* oder einem *absoluten* Pfadnamen an. Wenn Sie einen relativen Pfadnamen verwenden (das heißt, einen Pfad, der nicht mit *^/* beginnt), können Sie die Dateien später in einem beliebigen Verzeichnis wiederherstellen. Bei einem absoluten Pfadnamen (also einem vollqualifizierten Pfadnamen, der mit *^/* beginnt) stellt **tar** die kopierten Dateien *ausschließlich* in dem Verzeichnis wieder her, von dem Sie sie ursprünglich kopiert haben. Vor der eigentlichen Datei stellt **tar** sogar alle Verzeichnisse wieder her, die zu der Datei führen (selbst wenn es diese Verzeichnisse erst erstellen muß).

Hinweis: Es wird dringend empfohlen, daß Sie relative Pfadnamen verwenden, wenn Sie **tar**-Dateien erstellen. Dadurch sorgen Sie bei der Wiederherstellung für größtmögliche Flexibilität.

Beispiele für die Verwendung von *tar*

1. Das folgende Beispiel veranschaulicht die Anwendung von **tar**. Beachten Sie, daß Sie in der Befehlszeile mehrere Verzeichnisse oder Dateien angeben können. Wechseln Sie zunächst in das Verzeichnis über, von dem Sie die Datei oder das Verzeichnis kopieren möchten:

```
cd /
tar c ./etc ./adm
```

Dieser Befehl kopiert die Verzeichnisse **./etc** und **./adm** auf das voreingestellte Gerät **/dev/rxt0**. Beachten

Sie, daß relative Pfadnamen (beginnend mit `^./^`) verwendet werden, um die zu kopierenden Verzeichnisse zu identifizieren.

2. Möchten Sie den gesamten Inhalt der Archivkopie wiederherstellen, wechseln Sie in das gewünschte Verzeichnis und geben Sie folgendes ein:

```
tar xp
```

Da Sie für die Backup-Kopie einen relativen Pfadnamen verwendet haben, stellt **tar** das Backup in Ihrem aktuellen Arbeitsverzeichnis wieder her. Die Option **p** bewahrt die Berechtigungen der Datei. Beachten Sie, daß **tar** zwar versucht, die korrekten Berechtigungen wiederherzustellen. Dies kann jedoch nicht garantiert werden.

Weitere Informationen finden Sie im UNIX-Handbuch unter **tar**.

Das Programm `^cpio^`

Der Name **cpio** steht für `^copy input to output^` (Eingabe zu Ausgabe kopieren). Dieser Befehl kopiert Dateien in und aus einem **cpio**-Archiv. Er verwendet die Standard-Eingabe als Quelle für Dateinamen und die Standard-Ausgabe als Archivausgabe. Das bedeutet, daß Sie **cpio** mit standardmäßigen UNIX-Befehlen (wie **find** oder **ls**) kombinieren können, um eine Vielzahl spezifischer Backups anzulegen. Sie können z. B. **find** und **cpio** kombinieren und damit ein Backup aller Dateien eines bestimmten Benutzers erstellen, deren letzte Änderung länger als sechs Wochen zurückliegt.

Vorteile von `^cpio^`:

- Kann einzelne Dateien kopieren und wiederherstellen.
- Kann Zuwachs-Backups durchführen.
- Kann verwendet werden, um Dateien nach beliebigen Kriterien zu kopieren (Alter, Eigentümer, Größe usw.).

Nachteile von `^cpio^`:

- Hat eine kryptische Befehlssyntax.
- Nutzt Speicherplatz nicht effektiv.

- Beschränkt Pfadnamen auf 128 Zeichen.
- Kann nicht für Backups von Verzeichnissen oder Dateien mit symbolischen Verweisen verwendet werden.

Das Programm **cpio** verwendet für Backups zwei Optionen:

- **-o** kopiert in das Archiv (zur Erstellung eines Backups).
- **-i** kopiert aus dem Archiv (zur Wiederherstellung eines Backups).

Jede dieser Primäroptionen kann mit zusätzlichen Optionen kombiniert werden, um die Wirkung von **cpio** weiter zu modifizieren. Im UNIX-Handbuch finden Sie unter **cpio** eine vollständige Beschreibung dieser Optionen.

Im Gegensatz zu **dump** oder **tar** wird der Befehl **cpio** nicht alleine verwendet, um eine Backup-Kopie zu erstellen. Statt dessen erhält der Befehl von einem anderen UNIX-Befehl (normalerweise **find** oder **ls**) eine Liste der Dateien, die kopiert werden sollen. Die Argumente, die Sie mit diesen UNIX-Befehlen zusammen verwenden, bestimmen, welche Verzeichnisse und Dateien **cpio** kopiert.

Beispiele für die Verwendung von **cpio**

1. Im folgenden Beispiel kopiert **cpio -o** alle Dateien des aktuellen Verzeichnisses in ein Archiv. Beachten Sie, wie aus der Ausgabe des Befehls **ls** die Eingabe für **cpio** wird:

```
ls | cpio -oB > /dev/rxt0
```

Wichtig: Wenn Sie die Ausgabe an ein Bandlaufwerk senden (wie etwa **/dev/rxt0**), müssen Sie die Option **B** verwenden, mit der **cpio** Datensätze von 5.120 Byte verwendet. Diese Option verwenden Sie anschließend auch zur Wiederherstellung.

2. Der nächste Befehl sucht alle Dateien in **/NeXTLibrary/Documentation**, für die kein NFS-mount durchgeführt wurde, und archiviert sie anschließend:

```
cd /NeXTLibrary/Documentation
find . -depth -fstype 4.3 -prune -print | cpio -oB > /dev/rxt0
```

Im UNIX-Handbuch finden Sie weitere Informationen zu **find**.

3. Möchten Sie ein ganzes Archiv wiederherstellen, geben Sie etwa folgenden Befehl ein:

```
cpio -idB < /dev/rxt0
```

Die Option **-d** weist **cpio** an, während der Wiederherstellung alle erforderlichen Verzeichnisse zu erstellen.

4. Der folgende Befehl führt den Inhalt eines Archivs im vollständigen Format auf.

```
cpio -ivt < /myfile
```

Dieser Befehl kombiniert **cpio -i** mit der Option **-t**, die ein Inhaltsverzeichnis des Eingabearchivs erstellt, und mit der Option **-v** (die die Dateinamen im vollständigen Format auflistet, ähnlich der Ausgabe des Befehls **ls -l**).

Weitere Informationen über **cpio** finden Sie im UNIX-Handbuch unter **cpio**, **find** und **ls**.

Allgemeine Tips

Wenn Sie Backups durchführen, liegt die größte Gefahr beim Potential für menschliche Fehler. Wenn Sie nicht aufpassen, kann es leicht passieren, daß Sie wichtige Daten überschreiben. Ebenso kann man leicht vergessen, ein Backup zu machen, was oft schwerwiegende Folgen hat. Die folgenden Richtlinien helfen Ihnen dabei, Probleme zu vermeiden:

- Sie sollten auf jeden Fall einen systematischen Backup-Plan haben und sich strikt daran halten.
- Bevor Sie Daten wiederherstellen, sollten Sie sich vergewissern, daß beim Zurückspielen auf die Platte keine wichtigen Dateien überschrieben werden können. Die riskantesten Befehle in dieser Beziehung sind **restore r** (da **restore** immer in das aktuelle Verzeichnis kopiert), **rrestore r**, **tar x** und **cpio -i**.
- Achten Sie bei Backup-Programmen wie **dump** und **restore** besonders auf Tippfehler, da Sie leicht Ihre Platte überschreiben können. Führen Sie möglichst keine Backups aus, wenn Sie als ^a**root**^o angemeldet sind.
- Eventuell führen Sie ein vollständiges Backup durch, das Sie später auf jeden Fall wiederherstellen möchten

• Sie möchten beispielsweise eine Platte ersetzen und machen zu diesem Zweck erst ein Backup. In diesem Fall sollten Sie am besten zwei Kopien anlegen. Dadurch vermeiden Sie Datenverlust, falls die Daten auf einer Kopie zerstört werden. Außerdem sollten Sie überprüfen, ob das Backup erfolgreich durchgeführt wurde. Lesen Sie dazu den Inhalt des Backup-Bandes oder der Backup-Platte.

- Beschriften Sie Ihre Backup-Kopien immer sorgfältig und bewahren Sie sie an einem sicheren Ort auf.
- Wenn Benutzer wichtige Daten auf lokalen Platten speichern, die für die üblichen Backup-Prozeduren des Servers nicht verfügbar sind, sollten Sie oder die Benutzer unbedingt regelmäßig Backups für diese Daten durchführen.
- Bevor Sie ein Backup durchführen, muß der Schreibschutz des entsprechenden Datenträgers deaktiviert werden.

Störungsbeseitigung

Es folgt eine Liste mit Fehlermeldungen, die bei den einzelnen Backup-Programmen erscheinen könnten.

Meldungen von `adump` und `ardump`

Folgende Meldung erscheint, wenn kein Bandlaufwerk auf dem Computer vorhanden ist, auf dem Sie die Daten speichern möchten, oder falls Sie das Bandlaufwerk nicht richtig angegeben haben:

```
DUMP:NEEDS ATTENTION: Cannot open tape. Do you want to retry the open?
```

Wenn im verwendeten Computer die Datei `/etc/fstab` fehlt, erscheint folgende Meldung:

```
DUMP: Can't open /etc/fstab for dump table information.
```

Meldungen von `ardump` und `arrestore`

Wenn Sie weder für den Band-Server noch für das System, von dem Sie ein Backup anlegen, Remote-Zugriffsrechte haben, erhalten Sie eine Meldung wie die folgende:

```
Permission denied.
```

Sollten Sie eine solche Meldung erhalten, nehmen Sie die entsprechenden Einträge in den Dateien **/etc/hosts** der Systeme vor, die Ihnen den Zugriff verweigern.

Meldungen von **tar**

Eine Meldung ähnlich der folgenden erscheint, wenn kein Bandlaufwerk auf dem Computer vorhanden ist, auf dem Sie die Daten speichern möchten (mit **tar -c**), oder falls Sie das Bandlaufwerk fehlerhaft angegeben haben:

```
/dev/rxt0: No such tape or address.
```

Wenn Sie kein **root**-Zugriffsrecht haben oder nicht zur Gruppe **operator** gehören, erscheint eine Meldung wie die folgende:

```
Cannot open /dev/rsd0a
```

Meldungen von **cpio**

Das Programm **cpio** kann Pfadnamen mit mehr als 128 Zeichen nicht richtig verarbeiten. Haben die zu kopierenden Verzeichnisse und Dateien längere Namen, antwortet **cpio** mit einer Meldung wie etwa der folgenden:

```
<toolongpathname>?  
2936 blocks
```

Der Befehl **find** wird häufig zusammen mit **cpio** verwendet. Die Option **-ncpio** kann jedoch nicht mit **find** zusammen angewendet werden, ansonsten erhalten Sie eine Meldung wie die folgende:

```
find: bad option <-ncpio>
```

