

smart

System-Manager für
Archivierung, Restaurierung und
Transport von Daten

Andreas Koppenhöfer

23. Dezember 1996

Zusammenfassung

Diese Diplomarbeit diskutiert Aufgaben und Probleme einer zentralen Datensicherung und bietet dafür Lösungen an.

Es werden verschiedene Datensicherungsmethoden mit ihren Vor- und Nachteilen besprochen. Neben der Auswahl geeigneter Sicherungsmethoden und der Optimierung verschiedener Parameter, enthält diese Dokumentation auch Informationen über Dimensionierung und den praktischen Betrieb einer zentralen Datensicherung.

Als eine Lösung der Aufgaben einer zentralen Datensicherung wird *smart* vorgestellt: "*System-Manager für Archivierung, Restaurierung und Transport*". *smart* ist ein System bestehend aus Konzepten und Software zur Durchführung von regelmäßigen, automatischen Datensicherungen einer großen Anzahl von Rechnern eines Intranet (großes lokales Netzwerk), in dem vorwiegend UNIX eingesetzt wird. Dabei kann die Datensicherung mehrerer Rechner parallel auf einem oder mehreren zentralen Bandlaufwerken erfolgen, wobei auch Bandwechselroboter (changer devices) unterstützt werden.

Inhaltsverzeichnis

Zusammenfassung

1	Einleitung	1
2	Anforderungen	3
3	Zentrale Datensicherung	5
3.1	Relevante Eigenschaften von Daten bei einer Sicherung . . .	5
3.2	Datensicherungsmethoden	7
3.3	Anpassung der gewählten Sicherungsmethode	13
3.4	Organisation von Datensicherungen	16
3.5	Definition verwendeter Begriffe	17
3.6	Probleme und Dimensionierung	18
4	Datensicherung mit <i>smart</i>	23
4.1	Device Manager	25
4.1.1	dmui – Device Manager User Interface	26
4.1.2	xdmui – X11-based Device Manager User Interface .	26
4.1.3	dmcp – Device Manager Copy	26
4.2	Volume Manager	27
4.2.1	Verwendung eines Cache	28
4.2.2	Fragmente	31
4.2.3	Sicherungsintervall	31
4.2.4	Zeitrahen	32
4.2.5	Kombination von Zeitrahen und Intervall	33
4.2.6	Priorität	37

4.2.7	Verbinden mehrerer Datensicherungen	38
4.2.8	Gruppen	39
4.2.9	Aufbewahrungsfristen	39
4.2.10	Datensicherungsprogramme	40
4.3	<i>smart</i> -Benutzerschnittstelle	44
5	Restaurierung	47
5.1	Restaurierung mit <i>smart</i>	49
5.1.1	Restaurierung einzelner Dateien/Verzeichnisse	50
5.1.2	Restaurierung größerer Datenbestände	51
5.2	Restaurierung ohne <i>smart</i>	51
5.2.1	Zerstörte <i>smart</i> -Datenbank	51
5.2.2	Rekonstruktion von Inhaltsverzeichnissen	51
5.2.3	Restaurierung über Device Manager	52
5.3	Schwierigkeiten einer Restaurierung	53
5.3.1	Hardware	53
5.3.2	Software	56
5.3.3	Suche im Archiv	57
6	Aufbau und Betrieb von <i>smart</i>	59
6.1	Device Manager	59
6.1.1	Implementierung	59
6.1.2	Installation	59
6.1.3	Konfiguration	60
6.2	Volume Manager	63
6.2.1	Implementierung	63
6.2.2	Installation	65
6.2.3	Konfiguration	65
6.3	<i>smart</i> -Konfiguration	69
6.3.1	Benutzerverwaltung	69
6.3.2	Zeitraumen	70
6.3.3	Access-Control-Listen	71
6.3.4	Cache	71

6.3.5	Bandlaufwerke	73
6.3.6	Datensicherungen	73
7	Ausblick	77
	Anhang	
A	Device Manager	79
A.1	Syntaxdiagramme	80
A.1.1	Allgemeine Syntaxelemente	81
A.1.2	DM-Objektnamen	83
A.1.3	DM-Kommandos	85
A.1.4	Syntax Konfigurationsfiles	93
A.2	Benutzerinterface <code>xdmui</code>	100
A.3	Benutzerinterface <code>dmcp</code>	100
B	Formular “Anforderung einer Restaurierung”	105
C	Statistiken	107
C.1	Änderungshäufigkeit von Daten	107
C.2	Betriebsstatistik	108
D	Bezugsquellen	109
D.1	Datenbank	109
D.2	Perl und Perl-Module	110
D.3	WWW-Server	111
	Glossar	113
	Literaturverzeichnis	119
	Index	121

Abbildungsverzeichnis

3.1	Änderungshäufigkeit von Daten	6
3.2	Sicherungsumfang in Abhängigkeit vom Schwellwert für verschiedene Sicherungssequenzen	14
3.3	Sicherungsumfang in Abhängigkeit vom Schwellwert für verschiedene Datentypen	15
3.4	Datensicherung Betriebsstatistik – Vergleich von gesicherter Datenmenge und Zeitbedarf	19
4.1	Struktur der zentralen Datensicherung	23
4.2	Datensicherung eines Filesystems auf ein Bandlaufwerk	28
4.3	Datensicherung mehrerer Filesysteme auf eine entsprechende Zahl von Bandlaufwerken	28
4.4	Datensicherung mehrerer Filesysteme auf ein Bandlaufwerk	29
4.5	Aufspaltung von Datencontainern (Volumes) in kleinere Teile (Fragmente)	29
4.6	Mögliche Reihenfolge von Fragmenten auf einem Datensicherungsmedium	29
4.7	Zeitpunkt einer Datensicherung festgelegt durch Zeitrahmen	33
4.8	Berechnung eines Sicherungszeitpunkts aus Zeitpunkt der letzten Sicherung, Zeitrahmen und Intervall	34
4.9	Berechnung eines Sicherungszeitpunkts aus Anfangszeit eines Zeitrahmens und Intervall	34
4.10	Berechnung eines Sicherungszeitpunkts; mehrere Intervalle innerhalb eines Zeitrahmens	35
4.11	Berechnung eines Sicherungszeitpunkts; Zeit zwischen zwei Sicherungen kann zu kurz werden	35
4.12	Berechnung eines Sicherungszeitpunkts; Mindestzeit zwischen Sicherungen als Verzögerung eines Intervalls	36

4.13 Berechnung eines Sicherungszeitpunkts; mehrere Zeiträumen innerhalb eines Intervalls	36
4.14 Zentrale Datensicherung mit <i>smart</i> – beteiligte Instanzen . .	44
A.1 Device Manager – Objekttypen	80
A.2 <i>xdmui</i> : X11-based Device Manager User Interface, Hauptfenster	101

Kapitel 1

Einleitung

Daten sind wertvoll. Es kostet Zeit und Geld alle für eine Unternehmung oder Aufgabe benötigten Informationen zusammenzutragen und die zur Verarbeitung notwendigen Programme zu schreiben. Deshalb ist es wichtig, daß diese Daten nicht versehentlich gelöscht, verändert oder zerstört werden. Daten können verloren gehen durch Defekte an Computern oder Datenspeichern, Diebstahl, Sabotage oder Katastrophen wie z. B. Feuer.

Datensicherung und Datenschutz sind Themen, die diese Gefahren behandeln. Datensicherung im Sinne dieser Diplomarbeit besteht aus Methoden und Aktionen, mit denen sich Daten nach deren Löschung oder Verfälschung (im folgenden kurz Datenausfall oder Datenverlust genannt) wieder in ihrem ursprünglichen Zustand herstellen, restaurieren lassen. Dagegen kann Datenschutz präventiv gegen sabotagebedingten Datenausfall oder -verfälschung schützen. Diese Aspekte des Datenschutzes sollen hier jedoch nicht weiter betrachtet werden. Sie werden u. a. in [BA 94] und [IT 96] behandelt.

Eine Datensicherung erstellt Kopien der zu sichernden Daten, die nach einem Datenausfall zur Restaurierung verwendet werden können. Vom Verfahren der Kopienherstellung, sowie von Beschaffenheit und Lagerung der Kopien hängt es ab, ob Daten nach einem Ausfall restaurierbar sind. Restaurierbarkeit beruht also auf der Qualität der Datensicherung. Hohe Qualität bedeutet, daß jeder Zustand der Daten eines beliebigen zurückliegenden Zeitpunktes wiederhergestellt werden kann. Besitzt die Datensicherung eine niedrige Qualität, so lassen sich zwar die Zustände der Originaldaten bestimmter Zeitpunkte restaurieren, jedoch bleiben alle Zwischenstadien der Daten, die zwischen zwei aufeinanderfolgenden Zeitpunkten bestanden, verloren.

Vor der Entscheidung, welche Methode oder Konzept zur Datensicherung eingesetzt werden soll, müssen deren Ziele festgelegt werden. Hohe Qua-

lität in der Datensicherung verlangt einen hohen Aufwand an Personal und Material. Diese Tatsache gilt für die Sicherung eines einzelnen Rechners wie für die Sicherung aller Rechner eines großen lokalen Netzwerks (Intranet).

In einem Intranet läßt sich zur Datensicherung rationell ein zentralisiertes Sicherungssystem einsetzen, welches die Datenbestände aller angeschlossener Rechner auf einem oder mehreren zentralen Massenspeichern sichert. Aufgabe dieser Diplomarbeit war die Realisierung einer solchen zentralen Datensicherung.

Kapitel 2

Anforderungen

Vor Beginn dieser Diplomarbeit wurden Anforderungen für ein neues zentrales Datensicherungssystem aufgestellt. Die Aufgabe bestand aus Entwicklung und Implementierung eines Konzepts zur Durchführung einer zentralen Datensicherung. Sie wurde unter dem vorläufigen Titel *“Automatisierte Datensicherung (Backup)”* wie folgt formuliert.

“In einem heterogenen Computernetzwerk soll ein zentrales Datensicherungssystem eingesetzt werden. Dabei müssen die lokalen Daten einer größeren Anzahl unterschiedlicher Rechner auf einem (mehreren) zentralen Massenspeicher gesichert werden. Auf den beteiligten Rechnern sind verschiedene UNIX-Betriebssysteme installiert. Sie sind untereinander mit einem lokalen Netzwerk verbunden.

Für das bisher an der Fakultät Informatik eingesetzte Datensicherungsverfahren soll ein leistungsfähiger und flexibler Ersatz geschaffen werden. Folgende Leistungen werden gewünscht:

- *Es sollen große Datenmengen von einer Vielzahl unterschiedlicher Rechner (Clients) auf einem oder mehreren Massenspeichermedien gesichert werden.*
- *Die Daten sollen mit Hilfe eines bereits existierenden Systemdienstes transportiert werden ([DM], Studienarbeit 1298: ‘Implementierung eines Datentransportservices in einem heterogenen Computer-Netzwerk’).*
- *Die Datensicherung soll automatisch ablaufen können, d. h. es soll über einen bestimmten Zeitraum (Nacht, Wochenende, Feiertage) kein Operator benötigt werden.*
- *Ein Daemon soll als zentrale Instanz nach vorgegebenen Regeln entscheiden, welche Daten, wann und wie gesichert werden. Auch die*

Wahl des Massenspeichers bei automatischem Betrieb soll dieser Daemon treffen.

- *Die Regeln sollen Angaben enthalten über die Art der Daten und wie sie zu sichern sind, Zeitfenster und Prioritäten.*
- *Über ein Operator-Interface sollen privilegierte Operationen möglich sein, z. B. Änderung der Konfiguration und Datensicherungsregeln, manuelle Datensicherung, Abfragen von Informationen und Status der Datensicherung aller Clients, Wiederfinden gesicherter Daten anhand geeigneter Inhaltsverzeichnisse und Datenrestaurierung.*
- *Informationen über den Status der Datensicherung einzelner Clients sollen für die Administratoren der jeweiligen Rechner verfügbar sein.*
- *Unberechtigte Zugriffe auf gesicherte oder zu sichernde Daten müssen wirksam verhindert werden. Ebenso dürfen die Daten der Clients in ihrem Inhalt nicht verändert werden.*
- *Störungen oder Fehler beim Datentransport, eines Massenspeichers, Clients oder des Netzwerks sollen möglichst keine Auswirkung auf andere Datensicherungsaufträge haben.*
- *Die zur Verfügung stehenden Ressourcen (Massenspeicher, Übertragungskapazitäten, CPU-Leistung) sollen möglichst ökonomisch ausgenutzt werden.*
- *Es sollen Statistiken über Datensicherung und -Restaurierung generiert werden, die eine Abrechnung der Kosten nach Abteilung oder Client ermöglichen."*

Basierend auf diesen Anforderungen wurde eine Software entwickelt, die in den folgenden Kapiteln mit dem Namen *smart* bezeichnet wird.

Kapitel 3

Zentrale Datensicherung

Für eine Auswahl und Entwicklung von Methoden und Konzepten einer zentrale Datensicherung ist eine Betrachtung verschiedener Eigenschaften der zu sichernden Daten, Formen des Datenausfalls und möglicher Sicherungsmethoden nötig.

3.1 Relevante Eigenschaften von Daten bei einer Sicherung

Die folgende Aufstellung zählt einige besondere Eigenschaften der Daten auf, welche bei einer Datensicherung zu beachten sind.

1. Anzahl und Größe der Daten;
2. Zeit, die für eine Sicherung der Daten benötigt wird;
3. Wert der Daten;
4. Änderungsdatum der Daten.

Die Eigenschaften 1 und 2 werden im Abschnitt 3.2 genauer abgehandelt. Es folgen die Erläuterungen zu Eigenschaften 3 und 4.

Je wertvoller die Daten sind, desto besser müssen sie gegen Ausfall gesichert werden. Rechtliche Bestimmungen gehören ebenfalls zum Wert der Daten. So müssen z. B. bestimmte Unterlagen eines Unternehmens für mehrere Jahre aufbewahrt, archiviert werden.

Änderungsdatum ist der Zeitpunkt, an dem die zu sichernden Daten zuletzt verändert wurden. Wie im nächsten Abschnitt 3.2 deutlich wird, ermöglicht diese Zeitangabe eine Reduzierung des Umfangs einer Datensicherung.

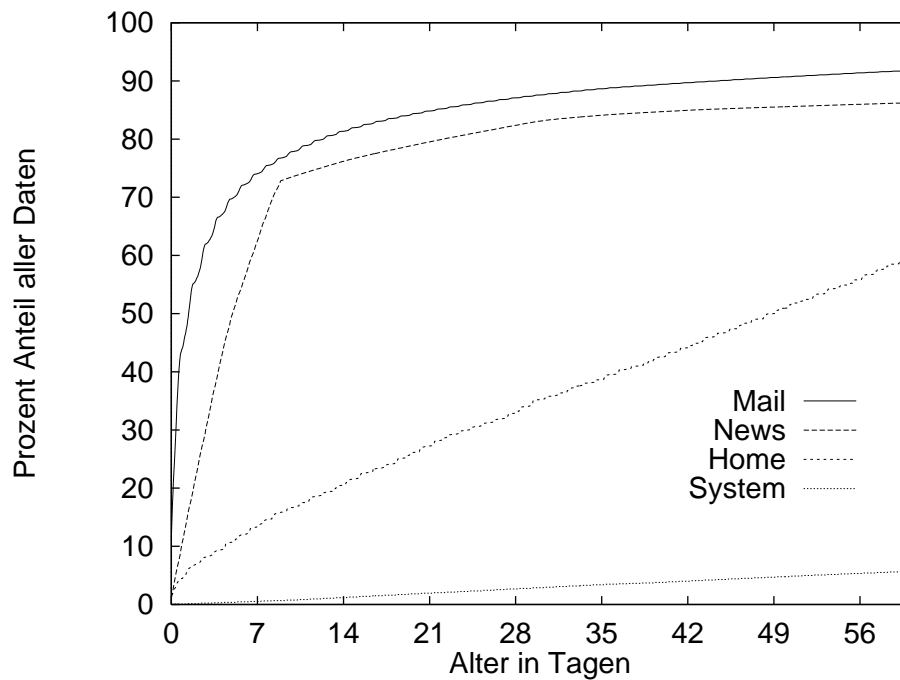


Abbildung 3.1: Änderungshäufigkeit von Daten

Die Änderungshäufigkeit von Daten zeigt eine statistische Untersuchung (siehe Anhang C). Als ein Ergebnis aus dieser Untersuchung enthält Abbildung 3.1 die Änderungshäufigkeit in Abhängigkeit von der Zeit: eine Angabe von 13 % in sieben Tagen bedeutet, daß innerhalb dieser Zeit der angegebene Teil der Gesamtdaten verändert wurde.

Es lassen sich folgende Datentypen unterscheiden:

System: Daten, die das Betriebssystem eines Rechners bilden. Nach der einmaligen Betriebssysteminstallation ändern sich nur sehr wenige Daten.

Home: private Daten der Benutzer eines Rechners. Diese erfahren eine gleichmäßige Veränderung im Verlauf der Zeit.

News: Daten aus dem Usenet (siehe Glossar, Seite 117) werden in der Regel innerhalb kurzer Zeit gelöscht. Der in Abbildung 3.1 deutlich erkennbare Knick nach sieben Tagen deutet daraufhin, daß die meisten Usenet-Daten (Artikel) nur etwa eine Woche auf dem Usenet-Server gespeichert bleiben.

Mail: Daten, die persönliche Mitteilungen (electronic mail) einzelner Benutzern enthalten. Diese Daten werden wie auch Daten des Typs Home

gleichmäßig verändert, jedoch ist hier innerhalb kurzer Zeit ein großer Teil der Daten von den Veränderungen betroffen.

Die Abbildung 3.1 zeigt bei den Datentypen Home und Mail eine leichte Stufigkeit. Die Ursache dieser Stufen beruht auf der Tatsache, daß diese Daten während der Nachtstunden weniger Veränderungen erfahren als am Tag.

Nicht alle Arten von Daten lassen sich den hier gezeigten Typen zuordnen. Diese vier Typen treten jedoch am häufigsten auf. Allgemein läßt sich feststellen: werden innerhalb einer kurzen Zeitspanne von Stunden oder Tagen...

... nur ein kleiner Teil des Datenbestands verändert, entspricht deren Änderungshäufigkeit der des Datentyps Home oder System;

... viele Daten verändert, dann ist deren Typ mit Mail oder News vergleichbar.

Datenbestände der Typen Home und System enthalten meist einen großen Anteil Daten, der während eines langen Zeitraums (Wochen, Monate) nicht verändert wird.

3.2 Datensicherungsmethoden

Für die verschiedenen Situationen, in denen eine Datenrestaurierung nötig werden kann, sollen hier geeignete Methoden zur Sicherung der Daten gegenübergestellt werden. Dabei finden die im vorherigen Abschnitt 3.1 aufgezählten besonderen Eigenschaften von Daten Beachtung.

Situation A

Verlust einzelner Dateien oder Verzeichnisse

Beispiele für Situation A sind Benutzer, die versehentlich Daten aus ihrem Arbeitsbereich löschen. Ebenso gehören Fehlfunktionen von Hard- und Software dazu, die unmittelbar Datenausfälle zur Folge haben. In der Regel werden Ausfälle dieser Art schnell bemerkt. Zur Datensicherung genügt Methode 1:

Sicherungsmethode 1

Regelmäßiges Kopieren aller zu sichernden Daten an einen anderen Ort. Bei jeder Erstellung einer neuen Kopie, wird die zuvor erzeugte Kopie überschrieben.

Wird der Datenausfall nicht rechtzeitig bemerkt, dann kopiert die nächste Datensicherung bereits fehlerhafte Daten und überschreibt damit die letzte Kopie mit korrekten Daten. Methode 1 erlaubt es nicht einen Zustand vor der letzten Sicherung zu restaurieren.

Situation B

Der Vorgang zur Erstellung von Sicherungskopien Methode 1 wird vorzeitig abgebrochen durch z. B. Systemabsturz oder Stromausfall.

Nach Situation B ist in der Regel keine verwendbare Sicherungskopie vorhanden. Abhilfe für diesen Nachteil verspricht Methode 2:

Sicherungsmethode 2

Regelmäßiges Kopieren aller zu sichernden Daten an einen anderen Ort. Kopien aus zurückliegenden Sicherungen werden aufbewahrt.

Beispiel: ein Benutzer erstellt jeden Abend nach getaner Arbeit eine Kopie seiner Daten und legt sie unter Angabe des Datums in einem gesonderten Verzeichnis ab. Der Vorteil von Methode 2 ist die Restaurierbarkeit der Daten zurückliegender Zeitpunkte.

Diese Methode besitzt einen großen Nachteil: mit jeder Datensicherung wächst der Speicherplatzbedarf. Schon nach wenigen Arbeitstagen übersteigt der Platzbedarf der Kopien ein Vielfaches des Umfangs der eigentlichen Daten. Zur Reduzierung des Platzbedarfs bietet sich Methode 3 an:

Sicherungsmethode 3

Einmaliges Kopieren aller zu sichernden Daten (Gesamtsicherung). Danach werden nur noch einzelne Dateien gesichert (inkrementelle Sicherung), die seit der letzten Sicherung verändert oder neu erstellt wurden.

In der Regel wird während eines Arbeitstages nur ein kleiner Teil aller Daten verändert, so daß sich mit Hilfe der Methode 3 eine drastische Reduzierung des Datensicherungsumfangs erreichen läßt. Methode 3 eignet sich deshalb besonders für Daten vom Typ Home und System (siehe Abschnitt 3.1).

Ein bisher vernachlässigter Aspekt der Datensicherung heißt Zeit. Die Bearbeitungszeit eines Kopiervorgang ist u. a. vom Umfang der Daten abhängig. Je größer der Datenumfang, desto länger dauert die Datensicherung. Eine inkrementelle Sicherung mit Methode 3 beinhaltet also auch eine Beschleunigung der Datensicherung, da nur noch ein Teil der Daten gesichert werden muß.

Situation C

Durch einen Defekt gehen alle Daten eines Systems verloren.

Nach Eintreffen von Situation C ist zur Restaurierung zunächst der Stand der Daten aus der Gesamtsicherung wiederherzustellen. Danach muß der Reihe nach jede einzelne inkrementelle Sicherung restauriert werden, bis schließlich der Stand der Daten zum Zeitpunkt der letzten Sicherung erreicht wurde. Dies ist ein Nachteil der inkrementellen Sicherung nach Methode 3 aus folgenden Gründen:

- Die Restaurierung benötigt viel Zeit. Der Zeitbedarf steigt mit der Anzahl der inkrementellen Sicherungen.
- Alle Sicherungen müssen aufbewahrt werden, da eine Restaurierung nur möglich ist, wenn die komplette Sicherungssequenz von Gesamtsicherung bis zum gewünschten Zeitpunkt verfügbar ist.
- Für den Fall, daß eine einzelne inkrementelle Sicherung unbrauchbar ist, können die Zustände der Daten aus dieser und den nachfolgenden Sicherungen nicht mehr (vollständig) hergestellt werden, da die Änderungen eines Zeitabschnitts fehlen. Eine Datensicherung kann unbrauchbar werden, wenn z. B. das Band, auf dem die Sicherung gespeichert wurde, nicht mehr lesbar ist.

Damit die inkrementelle Sicherungsmethode praktisch einsetzbar wird, muß sie noch modifiziert werden.

Sicherungsmethode 4

Regelmäßiges Kopieren aller zu sichernden Daten (Gesamtsicherung) in größeren zeitlichen Abständen (z. B. einmal wöchentlich). Dazwischen werden nur noch einzelne Dateien gesichert (tägliche inkrementelle Sicherungen), die seit der letzten Sicherung (inkrementell oder gesamt) verändert oder neu erstellt wurden.

Vorteile von Methode 4 gegenüber 3 am Beispiel von wöchentlichen Gesamtsicherungen und täglichen inkrementellen Sicherungen:

- Falls eine einzelne inkrementelle Sicherung unbrauchbar ist, beeinträchtigt diese höchstens die Restaurierbarkeit der Daten bis zum Zeitpunkt der nächsten Gesamtsicherung.
- Falls eine einzelne Gesamtsicherung unbrauchbar ist, geht nur die Restaurierbarkeit der Daten über den Zeitraum zwischen zwei Gesamtsicherungen verloren (hier: eine Woche).
- Methode 4 ermöglicht eine sinnvolle Archivierung. Alte inkrementelle Sicherungen werden gelöscht, während Gesamtsicherungen aufbewahrt werden. Somit bleiben Daten weit zurückliegender Zeitpunkte zur Restaurierung verfügbar.

- Zur langfristigen Archivierung wählt man einige alte Gesamtsicherungen (z. B. erste Sicherung eines Monats) aus. Alle anderen werden gelöscht. Dadurch reduziert sich der Platzbedarf in Archiv erheblich.

Die folgende Sicherungsmethode 5 ist eine Variation von Nr. 4:

Sicherungsmethode 5

Regelmäßiges Kopieren aller zu sichernden Daten (Gesamtsicherung) in größeren zeitlichen Abständen (z. B. einmal wöchentlich). Dazwischen werden nur noch einzelne Dateien gesichert (tägliche inkrementelle Sicherungen), die seit der letzten Gesamtsicherung verändert oder neu erstellt wurden.

Im Unterschied zur Methode 4 werden bei Nr. 5 mit jeder inkrementellen Sicherung alle seit der letzten Gesamtsicherung veränderten Daten kopiert.

Der Vorteil von Methode 5 beruht auf der erhöhten Sicherheit. Falls eine einzelne inkrementelle Sicherung unbrauchbar ist, bleiben dennoch alle anderen Sicherungen restaurierbar.

Einen Nachteil stellt jedoch der erhöhte Sicherungsaufwand dar. Zwischen zwei Gesamtsicherungen werden immer alle geänderten Daten kopiert, einschließlich bereits gesicherter Daten aus unmittelbar vorhergegangenen inkrementellen Sicherungen.

Es erscheint sinnvoll alle Vor- und Nachteile der Methoden 4 und 5 gegeneinander abzuwägen. Eine Verbindung dieser beiden Methoden bietet einen Kompromiß zwischen Qualität und Aufwand.

Sicherungsmethode 6

Erstellen regelmäßiger Gesamtsicherungen wie bei Methode 4 und 5. Sie werden als Sicherungen der Stufe 0 (dump level 0) bezeichnet. Mit Zahlen von 1 bis 9 werden inkrementelle Sicherungen eingestuft¹. Die Einstufung legt den Sicherungsumfang fest: Bei jeder inkrementellen Sicherung werden alle, seit der letzten, kleiner eingestufteten Sicherung, kopiert.

Die Anwendung der Methode 6 läßt sich am besten an einem Beispiel zeigen:

Montag: Eine Gesamtsicherung kopiert alle Daten.

Dienstag: Inkrementelle Sicherung mit dump level 3 kopiert alle seit Montag veränderten Daten.

¹Die Beschränkung der dump level auf einstellige Zahlen stammt vom Datensicherungsprogramm "dump" des Betriebssystems BSD. Viele moderne UNIX-Systeme haben dieses Programm unter z. T. geänderten Namen übernommen.

Mittwoch: Inkrementelle Sicherung mit dump level 5 kopiert alle seit dem letzten nächstniederen dump level veränderten Daten, hier Level 3 vom Vortag. Also werden alle Änderungen eines Tages gesichert.

Donnerstag: Inkrementelle Sicherung mit dump level 3; der letzte nächstniedere dump level war hier die Gesamtsicherung vom Montag. Deshalb werden an diesem Tag alle seit Montag geänderten Daten gesichert. Eine Sicherung mit dump level 5 würde seit Dienstag geänderte Daten sichern. Ein höherer Level (z. B. Level 7) würde hier die Sicherung auf Daten eines Tages beschränken.

Allgemein gilt: je niedriger der dump level, desto mehr Daten werden gesichert, da eine Sicherung mit niedrigem Level auch diejenigen Daten kopiert, die von unmittelbar vorausgegangenen Sicherungen eines höheren oder gleichen Levels bereits gesichert wurden.

Die dump level wurden für das vorhergehende Beispiel willkürlich ausgewählt. Eine Folge von einzelnen Level wird als Sicherungssequenz (dump sequence) bezeichnet. Jede Sequenz muß mit einer Gesamtsicherung (Level 0) beginnen und endet vor der nächsten Gesamtsicherung. Die jeweiligen Level kommen der Reihenfolge nach zur Anwendung.

Die Sequenz aus dem Beispiel läßt sich als eine Folge von Ziffern schreiben: "0353". Ebenso können die bisher vorgestellten Sicherungsmethoden mit Angabe einer Sicherungssequenz charakterisiert werden:

Methode	Sequenz	Bemerkungen
1, 2	0	nur Gesamtsicherungen
3	0123456789...	darstellbar nur bis zu neunten inkrementellen Sicherung
4	0123456	wöchentliche Gesamtsicherung
5	0111111	wöchentliche Gesamtsicherung
6	0353	Beispiel zu Methode 6

Eine geeignete Wahl der Sicherungssequenz beeinflusst in hohem Maße den Umfang der Datensicherung und den Zeitbedarf zu ihrer Erstellung. Es folgen zwei Verbesserungen der zuletzt erwähnten Methoden.

Situation D

Alle zu sichernden Daten werden zwischen zwei Datensicherungen verändert, z. B. bei einer Neuinstallation des Betriebssystem eines Rechners.

In Situation D wirkt sich ein starres Befolgen einer vorgegebenen Sicherungssequenz negativ aus. Die nachfolgenden inkrementellen Sicherungen, würden den Umfang einer Gesamtsicherung erreichen.

Sicherungsmethode 7

Regelmäßige Datensicherung wie mit Methode 6 und einer vorgegebenen Sicherungssequenz. Übersteigt jedoch der Umfang der zu kopierenden Daten einen festgelegten Anteil (z. B. 70%) der Gesamtdaten (Schwellwert), wird die Sicherungssequenz wieder von vorne mit einer Gesamtsicherung begonnen.

Methode 7 reduziert bei günstiger Wahl des Schwellwerts in bestimmten Situationen den Umfang der Datensicherung.

Situation E

Keine Änderungen des Datenbestands seit der letzten Datensicherung.

Die zweite Verbesserung betrifft den Zeitpunkt der Datensicherung in Situation E. Solange keine Daten geändert wurden, besteht auch kein Bedarf an einer erneuten Datensicherung.

Sicherungsmethode 8

Regelmäßige Datensicherung wie mit Methode 6 und einer vorgegebenen Sicherungssequenz. Vor dem Start der jeweiligen Datensicherung wird geprüft, ob sich seit der letzten Sicherung Daten geändert haben. Falls Änderungen vorliegen, erfolgt keine Datensicherung; die Position innerhalb der Sequenz bleibt unverändert.

Bedingt durch die Änderungshäufigkeit von Daten kann Methode 8 zu einer erheblichen Verlängerung des Zeitraums zwischen zwei Gesamtsicherungen führen.

Sicherungsmethode 9

Regelmäßige Datensicherung nach Methode 6 mit vorgegebener Sicherungssequenz und den Verbesserungen aus Methode 7 (Abbrechen einer Sequenz bei Bedarf) und Methode 8 (Aussetzen der Sicherung bei unveränderten Daten).

Methode 9 liefert bei einer geeigneten Wahl von Sicherungssequenz und Schwellwert eine hohe Qualität der Datensicherung.

Neben den bisher dargestellten Methoden gibt es noch eine Reihe anderer, die sich von den bisher vorgestellten Methoden zum Teil grundlegend unterscheiden. Eine solche Methode ist das sogenannte Logging:

Sicherungsmethode 10

Bei der Änderung von Daten wird vom Betriebssystem oder Anwendungsprogramm die Differenz zwischen bisherigen und neuem Zustand ermittelt. Diese Differenz wird fortlaufend auf ein Sicherungsmedium geschrieben, protokolliert.

Methode 10 wird vorwiegend im Bereich der Datenbanken eingesetzt. Ihr Vorteil, die vollständige Restaurierbarkeit, wird von den Datenbanken selbst genutzt (z. B. Transaktionssicherheit). Nachteile von Methode 10 sind ein großer Speicherplatzbedarf sowie ein hoher Zeitbedarf zur Berechnung und Speicherung der Differenz. Deshalb ist diese Methode nicht für eine allgemeine Verwendung geeignet.

Zur Verwendung in einer zentralen Datensicherung erscheint die Sicherungsmethode 9 geeignet. Diese Methode bildet deshalb die Grundlage für nähere Untersuchungen und findet in der später vorgestellten *smart*-Software Verwendung.

3.3 Anpassung der gewählten Sicherungsmethode

Die im letzten Abschnitt 3.2 beschriebenen Sicherungsmethode 9 benötigt als Parameter die Angabe einer Sicherungssequenz und eines Schwellwerts. Diese sollten in Abhängigkeit von der Art der zu sichernden Daten gewählt werden (siehe Abschnitt 3.1, Änderungshäufigkeit von Daten). Anhand von gesammelten statistischen Daten (siehe Anhang C) können Auswirkungen verschiedener Schwellwerte und Sicherungssequenzen verglichen werden.

Änderungen des Sicherungsumfangs unter Verwendung verschiedener Schwellwerte und Sequenzen zeigt Abbildung 3.2. Diese Änderung bezieht sich jeweils auf die Sequenz 1 bei einem Schwellwert von 90 %. Ein Schwellwert von 100 % bedeutet, daß eine Sequenz nicht vorzeitig abgebrochen wird. Mit diesem Schwellwert verhält sich die Sicherungsmethode 9 wie Methode 8.

Für den Vergleich wurden folgende Sequenzen ausgewählt:

1. 05555 35555
Bisher zur zentralen Datensicherung an der Fakultät Informatik verwendete Sequenz.
2. 05555 35555 35555 35555
Verlängerung der Sequenz auf das doppelte. Dadurch werden Gesamtsicherungen seltener durchgeführt.
3. 057575757 357575757
Einschieben von zusätzlichen inkrementellen Sicherungen der Stufe 7 nach jeder Stufe 5 der ursprünglichen Sequenz.
4. 057575757 357575757 357575757
Verlängerung der Sequenz 3 um ein Drittel.

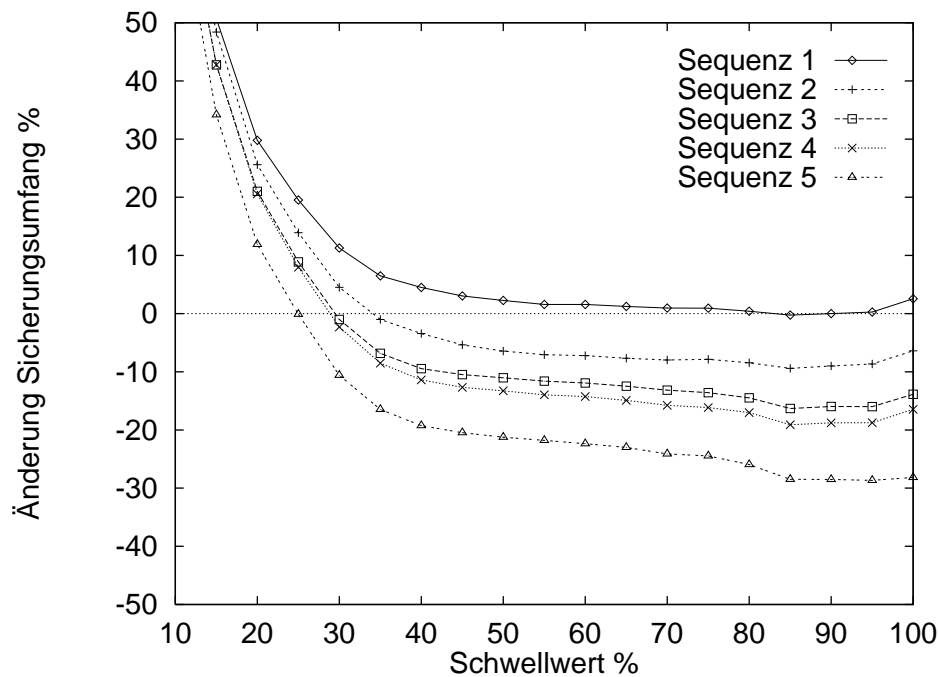


Abbildung 3.2: Sicherungsumfang in Abhängigkeit vom Schwellwert für verschiedene Sicherungssequenzen

5. 0123456789123456789

Bietet zur Datensicherung den geringsten Aufwand, jedoch verlangt eine Restaurierung bis zu zehn Arbeitsgänge.

Mit Sequenz 3 werden häufiger Gesamtsicherungen durchgeführt als mit 2. Dennoch bleibt der Umfang aller Sicherungen mit Sequenz 3 geringer.

Aus Abbildung 3.2 ist abzulesen, daß der Schwellwert größer als 50 % gewählt werden sollte. Sein optimaler Wert liegt hier bei 85 %.

Die gezeigten Werte gelten nur für die im Rahmen der statistischen Erhebung beobachteten Daten (Filesysteme). Die Wahl der Parameter hängt von der Änderungshäufigkeit der Daten ab. Abbildung 3.3 gibt für die Sequenz Nr. 4 einen Vergleich über die in Abschnitt 3.1 beschriebenen Datentypen.

Aus diesem Vergleich ergibt sich, daß für Daten, die nur wenig geändert werden (z. B. Daten des Typs System), eine günstige Wahl der Sicherungssequenz wichtig ist. Die Festlegung eines Schwellwerts ist hier von untergeordneter Bedeutung. Dagegen hat bei häufig veränderten Daten (z.B. Typ Mail) die Wahl des Schwellwerts mehr Einfluß auf den Datensicherungsumfang als eine Variation der Sicherungssequenz.

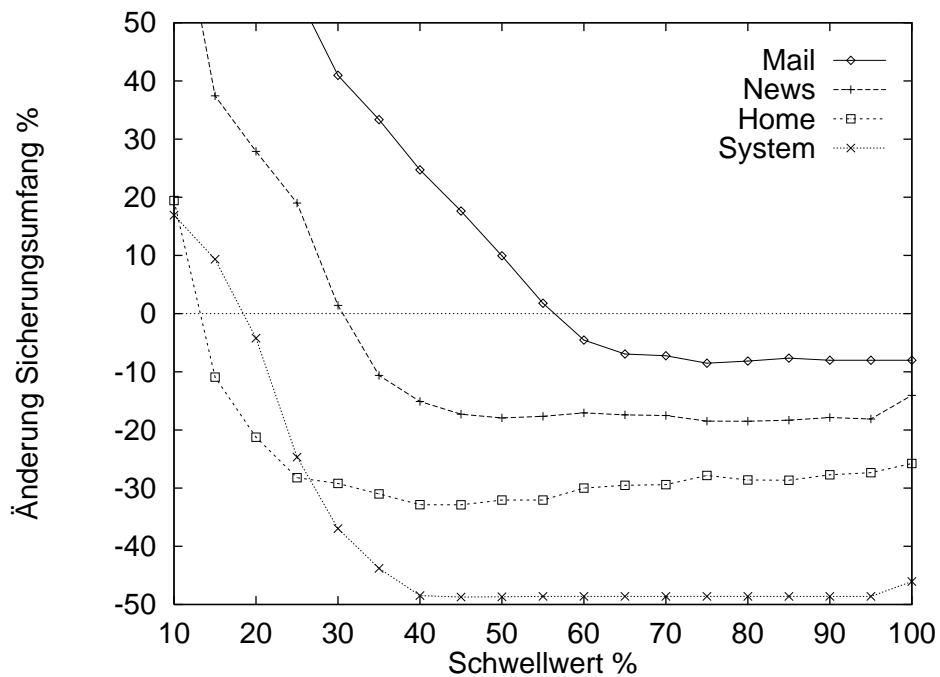


Abbildung 3.3: Sicherungsumfang in Abhängigkeit vom Schwellwert für verschiedene Datentypen

Für Daten des Typs Home erscheint ein niedriger Schwellwert von 40 bis 45 % günstig, während besonders für sich häufig ändernde Daten hohe Schwellwerte geeigneter sind.

Noch weitere Faktoren (Parameter) können den Umfang von Datensicherungen beeinflussen. Hierzu zählt das Sicherungsintervall. Ein Intervall gibt an, wie oft eine Sicherung durchgeführt wird. Wichtige Daten sollten täglich oder mehrmals täglich gesichert werden. Bei unwichtigen Daten (z. B. Daten des Betriebssystem) genügt meist eine wöchentliche Sicherung.

Die durchgeführten Untersuchungen zeigen deutlich, daß eine Trennung zu sichernder Daten in Bezug auf ihre Änderungshäufigkeit sehr sinnvoll ist. In der Regel werden Daten auf einer Festplatte in sogenannten Partitionen abgelegt. Übliche Datensicherungsprogramme wie z. B. `dump` arbeiten partitionsweise, d. h. sie sichern während eines Arbeitsgangs alle Daten einer Partition. Bei einer Aufteilung der Festplatte in verschiedene Partitionen sollten deshalb getrennte Bereiche für verschiedene Datentypen eingerichtet werden. Eine solche Trennung von z. B. Datentypen System und Home, bei geeigneter Wahl von Datensicherungsmethode, -Parameter und -Intervall, ermöglicht eine Reduzierung des Umfangs der Datensicherungen und damit eine Kostenersparnis.

3.4 Organisation von Datensicherungen

Datensicherung kann aus organisatorischer Sicht auf verschiedenen Ebenen eingesetzt werden.

Benutzerebene: Alle Benutzer eines Rechners sichern ihre eigenen Daten. Dabei müssen sie auf entsprechenden Speicherplatz für die Sicherungskopien zugreifen und ihn verwalten können. Dies setzt Kenntnisse von Methoden und Datensicherungsprogrammen bei jedem Benutzer voraus. Zudem ist vielen Benutzern die Notwendigkeit von Datensicherungen nicht bewußt. Als Folge werden Sicherungen von Benutzern nur selten oder überhaupt nicht durchgeführt. Gemeinsam verwendete Daten wie z. B. die des Betriebssystems werden hierbei nicht gesichert.

Rechnerebene: Der Administrator eines Rechners erstellt regelmäßig Datensicherungen für seinen Rechner. Jeder so gesicherte Rechner muß über ausreichend Speicherplatz für Sicherungskopien verfügen. Dieser Speicherplatz kann auf den Rechnern durch je ein Bandlaufwerk bereitgestellt werden. Im Vergleich zur Sicherung auf Benutzerebene benötigt nur ein Benutzer (Administrator) Datensicherungskenntnisse.

Abteilungsebene: Der Administrator einer Abteilung erstellt regelmäßig Datensicherungen aller Rechner seiner Abteilung. Der benötigte Speicherplatz kann z. B. durch ein Bandlaufwerk in jeder Abteilung bereitgestellt werden. Im Vergleich zur Sicherung auf Rechnerebene muß nicht jeder Rechner mit einem Bandlaufwerk ausgestattet sein.

Zentral: Sicherung aller Rechner eines Intranets auf einen oder mehreren zentralen Datenspeichern. Aufgrund des großen Sicherungsumfangs müssen hier z. B. schnelle Bandlaufwerke mit Bandwechselroboter eingesetzt werden.

In der dezentralen Datensicherung wird naturgemäß eine Vielzahl verschiedener Datensicherungsmethoden und Speichermedien eingesetzt. Die Austauschbarkeit von Sicherungen zwischen verschiedenen Rechnern ist aufgrund der Unterschiede fraglich. Dagegen ermöglicht eine zentrale Datensicherung den rationellen Gebrauch einheitlicher Methoden und Speichermedien. Mit Hilfe von Aufgaben- und Verantwortungskonzentration auf wenige Administratoren lassen sich Verfügbarkeit und Qualität der Datensicherung erhöhen (Stichwort: Kompetenzzentrum). Mehr über Verfügbarkeit und Qualität aus der Sicht des Managements kann in [BA 94, Chapter 6: Departmental Recovery] nachgelesen werden.

3.5 Definition verwendeter Begriffe

Die folgenden Kapitel verwenden einige Begriffe, deren spezielle Bedeutung wie folgt definiert werden.

Server: Zentraler Rechner bzw. Host, auf dem Software der zentralen Datensicherung läuft.

Client: Host, der zur regelmäßigen Datensicherung angemeldet ist. In der Regel ist ein Server zugleich auch Client.

Filesystem: Häufig verwendetes Synonym für zu sichernde Daten eines Client. Gemeint ist damit ein Teil des Datenbestands, der von den unter UNIX üblichen Datensicherungsprogrammen (z. B. `dump`) gemeinsam behandelt wird. Diese müssen aber nicht notwendigerweise als Filesystem organisiert sein.

Volume: Auf Clients erzeugen Datensicherungsprogramme einen Strom von Bytes, der die gesicherten Daten enthält. Dieser Datenstrom mit einem definiertem Anfang und Ende wird als Datencontainer oder Volume bezeichnet. Jedes Volume erhält eine eindeutige Nummer (Identifier, ID). Anhand dieser sogenannten Volumenummer oder Volume-ID lassen sich einzelne Datencontainer unterscheiden.

Fragment: Teil eines Volumes (Datencontainer); entsteht bei der Aufteilung eines Volumes in kleinere Stücke.

Gruppe: Clients und Benutzer können Gruppen zugeordnet werden. Über diese Gruppen lassen sich u. a. Zugriffsrechte bestimmen.

Administrator: Benutzer, der Dienste der zentralen Datensicherung nutzen kann. Zu diesen Diensten gehören u. a. Abruf von Informationen über die Datensicherung, Anforderung einer Restaurierung. Administratoren können Gruppen zugeordnet werden. Der Zugriff eines Administrators ist beschränkt auf Daten und Informationen der zu diesen Gruppen gehörenden Clients.

Operator: Administratoren der Gruppe "Operator" haben Zugriff auf alle gesicherten Daten und Inhaltsverzeichnisse, sowie deren Konfiguration. Operatoren sind für die Bestückung der Bandlaufwerke mit Medien zuständig und haben Zugang zum Bandarchiv.

Benutzerinterface: Programme zur Verwendung durch Administratoren.

Operatorinterface: Programme zur Verwendung durch Administratoren der Gruppe "Operator".

Console: Operatorinterface, über das die Software der zentralen Datensicherung Informationen und Fehlermeldungen ausgibt.

3.6 Probleme und Dimensionierung

Die zentrale Datensicherung muß verschiedene Probleme überwinden, die es zum Teil auf der Abteilungsebene gibt, nicht jedoch auf Rechner- oder Benutzerebene. Dazu gehören folgende Punkte:

- Für die zentrale Datensicherung müssen geeignete Datensicherungsmethoden gewählt werden, die auf den Clients einsetzbar sind. Die im Abschnitt 3.2 vorgestellten Datensicherungsmethoden 1 und 2 lassen sich sinnvoll nur zur Sicherung eines einzelnen Rechners einsetzen (dezentrale bzw. lokale Datensicherung). Für eine zentrale Datensicherung eignet sich dagegen die Sicherungsmethode 9: inkrementelle Sicherung.
- Durch eine Zentralisierung wird es für einen Administrator der zentralen Datensicherung schwierig, manuelle Arbeiten auf den Rechnern einer Abteilung durchzuführen, da diese in der Regel nicht dem direkten Verantwortungsbereich des Datensicherungsadministrators angehören. Arbeiten, die über die regelmäßige, automatisierte Datensicherung hinausgehen, können meist nur mit Hilfe des für den jeweiligen Client zuständigen Administrator erledigt werden. Eine Datenrestaurierung verlangt häufig solche manuellen Arbeiten. Das Verfahren einer Restaurierung bespricht Kapitel 5.
- Störungen einzelner zur Datensicherung angemeldeter Clients dürfen nicht die Datensicherung anderer beeinträchtigen. Typische Störungen sind Systemabsturz eines Clients oder Fehler bei der Datenübertragung.
- Verfügbare Datenspeicher (Bandlaufwerke) müssen bestmöglich ausgenutzt werden. Optimal ist das Schreiben und Lesen der Daten mit der Maximalgeschwindigkeit, die das Bandlaufwerk bieten kann. Der Grund hierfür wird nachfolgend erläutert.

Anhand einer Betriebsstatistik der Fakultät Informatik (siehe Anhang C) zeigt Abbildung 3.4 Datenübertragungsgeschwindigkeiten, die einzelne Rechner der zentralen Datensicherung maximal erreicht haben. Die eingezeichnete Grenze von 512 KByte² je Sekunde bestimmt das verwendete

²1 KByte = 1024 Byte = 2¹⁰ Byte

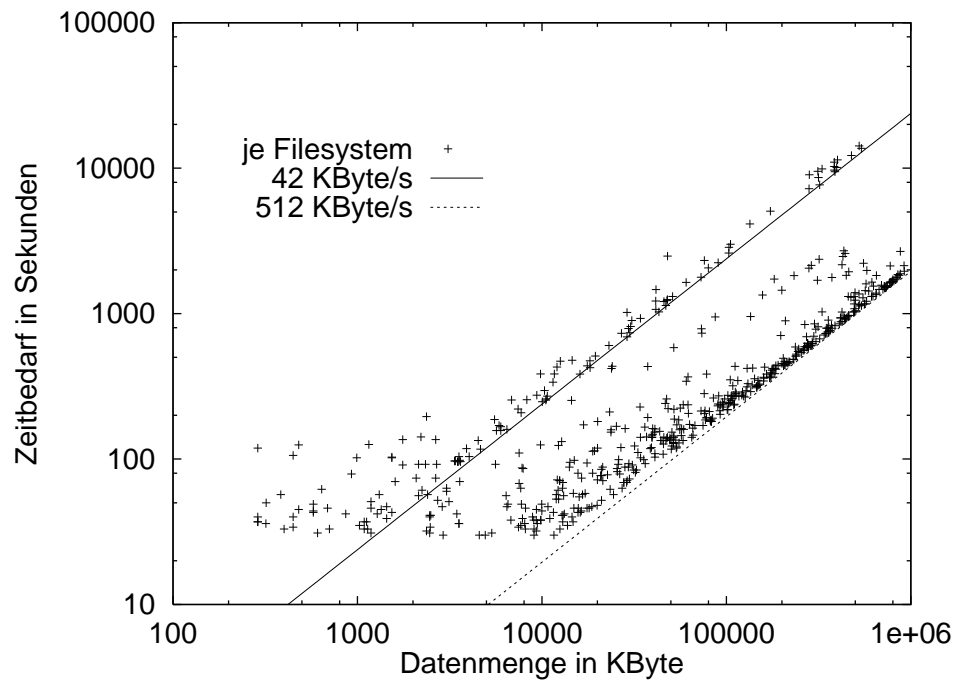


Abbildung 3.4: Datensicherung Betriebsstatistik – Vergleich von gesicherter Datenmenge und Zeitbedarf

Bandlaufwerk, das Daten nicht schneller annehmen kann. Die zweite Grenze wird von der Hardware einiger langsamer Rechner bestimmt, die während einer Sicherung Daten nicht schneller als mit 42 KByte je Sekunde an das Bandlaufwerk übertragen können. Die durchschnittliche Übertragungszeit aller Datensicherungen lag im Monat Oktober 1996 bei 351 KByte je Sekunde.

Bei zentraler Datensicherung können langsame Rechner ein Problem darstellen. Diese Schwierigkeit läßt sich mit Hilfe eines Beispiels aus der bereits erwähnten Betriebsstatistik verdeutlichen. Eine zentrale Datensicherung soll Daten aller Clients jede Nacht auf ein Bandlaufwerk sichern. Die Software führt dabei jede einzelne Sicherung der Reihe nach durch. Bei einem Datensicherungsdurchschnitt von 351 KByte/s und einer Laufzeit von 12 Stunden (20 Uhr bis 8 Uhr) können somit maximal 14.5 GByte³ in einer Nacht gesichert werden:

$$351 \frac{\text{KByte}}{\text{s}} \cdot 12\text{h} \cdot \frac{1 \text{ GByte}}{1024 \cdot 1024 \text{ KByte}} \cdot \frac{3600\text{s}}{1\text{h}} \approx 14.5 \text{ GByte}$$

Aus Sicherungssequenz und durchschnittlichem Sicherungsumfang läßt sich ein Datensicherungsbedarf der Clients bestimmen. Für die verwendete

³1 MByte = 1024 KByte = 2²⁰ Byte ; 1 GByte = 1024 MByte = 2³⁰ Byte

te Sequenz 05555 35555 weist die monatliche Betriebsstatistik⁴ folgende durchschnittliche Größen für den Umfang der Sicherungen nach:

dump level	Anzahl	Sicherungsumfang	
		Durchschnitt	je Sequenz
0	1	376 MByte	376 MByte
3	1	53 MByte	53 MByte
5	8	32 MByte	256 MByte
Gesamtumfang je Sequenz (10 Tage):		685 MByte	
Durchschnitt Umfang je Tag:		68.5 MByte	

Der durchschnittliche Umfang von 68.5 MByte je Sicherung ergibt bei 465 zu sichernden Filesystemen einen Sicherungsbedarf von 31 GByte täglich:

$$465 \cdot 68.5 \text{ MByte} \cdot \frac{1 \text{ GByte}}{1024 \text{ MByte}} \approx 31 \text{ GByte}$$

Im Vergleich dazu vermag die zur Sicherung eingesetzte Soft- und Hardware nur 14.5 GByte täglich zu sichern. Die hier dargestellte zentrale Datensicherung ist also stark unterdimensioniert. Als Folge können die Filesysteme der Clients nicht wie gewünscht täglich gesichert werden.

Als Abhilfe bieten sich folgende Maßnahmen an:

- Verbesserung der Software, so daß die Datensicherung mehrerer Clients parallel ablaufen kann. Diese Maßnahme verhindert, daß einzelne langsame Clients die zentrale Datensicherung "behindern" können. Parallele Sicherung setzt voraus, . . .
 - daß mehr als ein Bandlaufwerk zur Verfügung steht
 - oder daß die Software in der Lage ist, den Datenstrom mehrerer Clients gleichzeitig auf ein gemeinsames Bandlaufwerk zu schreiben (multiplexen).
- Einsatz eines schnelleren Bandlaufwerks.
- Veränderung von Parametern der eingesetzten Datensicherungsmethode (z. B. andere Sicherungssequenz), damit sich der Umfang der Datensicherungen reduziert.
- Vergrößerung des Sicherungsintervalls: nur wichtige Daten täglich sichern, für unwichtige reicht meist eine wöchentliche Sicherung.

⁴Stand November 1996

Tatsächlich sichert die Fakultät Informatik nur einen Teil der Daten täglich (Datentypen Home und Mail). Daten des Typs System sind zur zentralen Datensicherung nur mit einem wöchentlichen Intervall angemeldet. Insgesamt 70 % der zu sichernden Daten sind als Typ System klassifiziert. Der Rest von 30 % soll weiterhin täglich gesichert werden. Rechnerisch ergibt sich aus dem oben ermittelten Durchschnitt mit

$$68.5 \text{ MByte} \cdot \frac{30\% + \frac{70\%}{7 \text{ Tage}}}{100\%} = 27.4 \text{ MByte}$$

ein neuer Durchschnitt von 27.4 MByte Sicherungsumfang. Insgesamt beträgt damit der tägliche Sicherungsbedarf etwa 12.4 GByte:

$$465 \cdot 27.4 \text{ MByte} \cdot \frac{1 \text{ GByte}}{1024 \text{ MByte}} \approx 12.4 \text{ GByte}$$

Dieser reduzierte Sicherungsbedarf unterschreitet jedoch nur knapp die maximale Sicherungskapazität von 14.5 GByte. Laut Betriebsstatistik übersteigt der tatsächliche Sicherungsbedarf häufig den errechneten Durchschnittswert. Gründe dafür sind Verzögerungen auf Seiten der Clients während der Vorbereitung einer Datensicherung und Verwaltungsaufgaben der Zentrale (Führen von Inhaltsverzeichnissen). Als Folge davon können nicht immer alle Filesysteme innerhalb der vorgegebenen Zeit gesichert werden. Eine Verlängerung täglicher Sicherungsintervalle eines Teils der Daten auf wöchentliche reicht also nicht aus; es müssen weitere Maßnahmen zur Verbesserung der Sicherungsqualität getroffen werden.

Als Regel für die Dimensionierung von Bandlaufwerken zur Datensicherung wird in [LA 96] allgemein gefordert, "daß die gesamte zu sichernde Plattenkapazität auf ein Medium paßt". Für eine zentrale Datensicherung kann daraus abgeleitet werden, daß das System in der Lage sein muß, die gesamte zu sichernde Plattenkapazität innerhalb eines vorgegebenen Zeitraums (z. B. eine Nacht bzw. 12 Stunden) zu sichern. Da bei einer zentralen Datensicherung durchaus auch mehrere Bandlaufwerke und Bandwechselroboter Verwendung finden, ist eine Beschränkung auf ein Medium nicht sinnvoll.

Aus Erfahrungen und Problemen in Bezug auf bisher eingesetzter Software zur Datensicherung an der Fakultät Informatik entstand ein Bedarf nach besserer Sicherungssoftware. Eine solche Software soll nun vorgestellt werden.

Kapitel 4

Datensicherung mit *smart*

smart steht für eine Abkürzung aus den Anfangsbuchstaben von “*System-Manager für Archivierung, Restaurierung und Transport*”. *smart* stellt ein System dar, bestehend aus Konzepten und Software zur Durchführung von regelmäßigen, automatischen Datensicherungen einer großen Anzahl von Rechnern auf einem oder mehreren zentralen Bandlaufwerken.

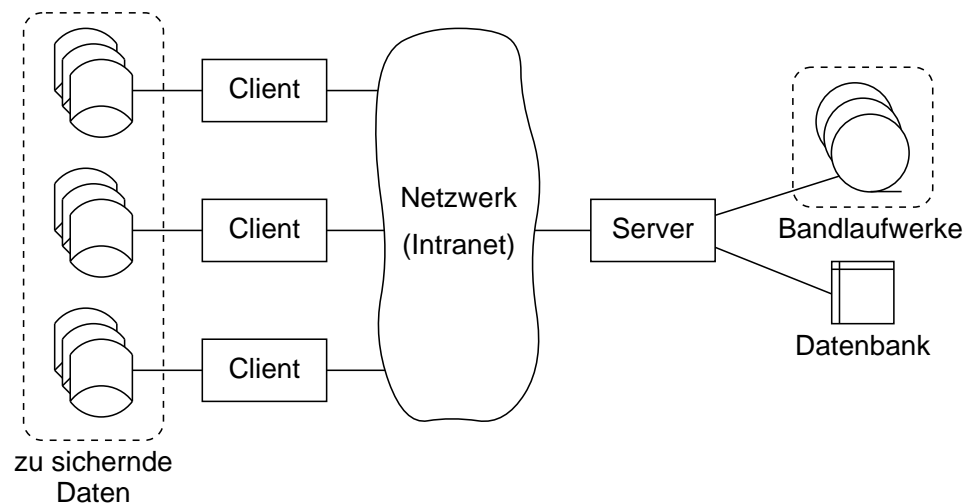


Abbildung 4.1: Struktur der zentralen Datensicherung

Die Struktur der zentralen Datensicherung unter *smart* zeigt Abbildung 4.1. Darin kontrolliert und steuert ein Server jede Datensicherung. Dieser Server verfügt über eine Datenbank, welche die zum Betrieb notwendigen Informationen enthält.

Zur Datensicherung entnimmt ein auf dem Server laufendes Programm Informationen aus der Datenbank. Es stellt fest, welche Daten zu sichern sind. Das Serverprogramm startet dazu über das lokale Netzwerk auf dem

jeweiligen Client ein Datensicherungsprogramm und übergibt diesem alle zur Sicherung notwendigen Informationen (Sicherungsmethode, Parameter, etc.). Das Sicherungsprogramm des Client beginnt daraufhin mit der "Verpackung" der zu sichernden Daten als Container und überträgt diesen Container über das Netzwerk an den Server¹. Der Server schreibt schließlich den empfangenen Datencontainer auf ein Speichermedium eines Bandlaufwerks. Unter Angabe einer für diesen Datencontainer vergebenen Nummer (Volume-ID) fügt er in das Inhaltsverzeichnis der Datenbank einen Eintrag mit Informationen über die Datensicherung ein.

Für die Restaurierung von Daten muß zunächst ein Operator den dazu benötigten Datencontainer im Inhaltsverzeichnis der Datenbank suchen. Über die gefundene Volume-ID des Datencontainers ermittelt der Server die Speichermedien, auf denen der Datencontainer abgelegt ist. Medien, die nicht bereits über ein Bandlaufwerk online verfügbar sind, fordert das Programm unter Angabe einer Mediennummer (Medienlabel) vom Operator an. Die Anforderungen erreichen die Operatoren als Mitteilung über eine spezielle Benutzerschnittstelle (Console). Ein Operator legt daraufhin die angeforderten Medien in die Bandlaufwerke ein und bestätigt ihre Verfügbarkeit. Nachdem die zur Restaurierung benötigten Medien (Bänder) online zugreifbar sind, beginnt der Server mit der Übertragung des Datencontainers an den vom Operator spezifizierten Client. Auf diesem Client kann schließlich der Datencontainer mit einem entsprechenden Programm "ausgepackt" werden.

Von *smart* verwendete Bandlaufwerke müssen nicht, wie in Abbildung 4.1 angedeutet, an denselben Host (Server) angeschlossen sein. Es ist z. B. möglich Bandlaufwerke verschiedener Clients mit in die zentrale Datensicherung einzubinden. Dasselbe gilt auch für die Datenbank. Sie darf auf jedem über das lokale Netzwerk erreichbaren Host installiert sein.

Eine zentrale Datensicherung läßt sich in verschiedene Aufgabenbereiche gliedern:

Backup: Starten von Datensicherungsprogrammen auf den Clients.

Transport: Übertragung der gesicherten Daten an einen zentralen Server mit Bandlaufwerk.

Devices: Schreiben von Daten auf Speichermedien der Bandlaufwerke und Bedienung von Bandwechselroboter; Lesen dieser Medien.

Verwaltung: Führen von Inhaltsverzeichnissen gesicherter Daten und Speichermedien.

¹Der Server delegiert die Aufgaben der Übertragung von Datencontainern und die Bedienung von Bandlaufwerken an einen Client. Deshalb werden Datencontainer genaue genommen immer nur zwischen zwei Clients übertragen.

Steuerung: Durchführung und Koordination der einzelnen Datensicherungen.

Archiv: Suchen und Finden von Einträgen gesicherter Daten in den Inhaltsverzeichnissen.

Restaurierung: Bereitstellen gesicherter Daten für eine Wiederherstellung und Übertragung dieser Daten vom Bandlaufwerk auf einen Client.

Konfiguration: Sammeln von Informationen in der Datenbank des Servers: zu sichernde Daten von Clients, Festlegung von Sicherungsmethoden und Parameter, Bereitstellen von Bandlaufwerken und Ressourcen zur Datensicherung.

Kontrolle: Prüfen von Protokollen durchgeführter Datensicherungen.

Statistiken: Erstellen von Datensicherungstatistiken; Kostenstellenrechnung auf Client- oder Abteilungsebene.

Software einer zentralen Datensicherung muß in der Lage sein, alle regelmäßig anfallenden Aufgaben automatisch und unbeaufsichtigt abzuarbeiten. Dazu gehören die zuerst genannten Bereiche Backup, Transport, Devices, Verwaltung sowie die Steuerung. Diese Aufgaben werden unter *smart* von einem sogenannten Volume Manager und mehreren Device Manager bearbeitet. Auf jedem Client ist ein Device Manager für die dort anfallenden Aufgaben zuständig, wogegen der Volume Manager für den Betrieb der Zentrale zuständig ist. In den Aufgabenbereichen Archiv, Restaurierung, Konfiguration, Kontrolle und Statistiken verfügt der Volume Manager über ein Benutzerinterface, mit dem ein Benutzer die entsprechenden Funktionen ansprechen kann.

Device Manager, Volume Manager und Benutzerinterface werden in den folgenden Abschnitten vorgestellt.

4.1 Device Manager

In den zuvor erwähnten Arbeitsbereichen Backup, Transport und Devices läßt sich Software einsetzen, die im Rahmen einer Studienarbeit entstanden ist. Die Software des Device Manager, die [DM] beschreibt, läßt sich auf vielen verschiedenen Betriebssystemen und Rechnerplattformen verwenden. Sie vereinfacht den Zugriff auf Bandlaufwerke und ermöglicht die Ansteuerung von Bandwechselrobotern. Der Device Manager kann mit den Eigenheiten der verschiedenen Betriebssystemen und der Hardware der Clients umgehen und bietet der zentralen Datensicherung eine einheitliche

Kommandoschnittstelle an. Zu den über die Kommandoschnittstelle verfügbaren Dienste gehören der Aufruf von Datensicherungsprogrammen, Übertragung von Datencontainer zwischen Client und Server, Bereitstellung eines Datencontainers zur Verarbeitung.

Der Device Manager besteht aus einem Programm, das bei Bedarf automatisch gestartet wird. Diese Kommandoschnittstelle und die Implementierung des Device Managers werden in [DM] ausführlich beschrieben. Eine kurze Zusammenfassung darüber ist im Anhang A aufgeführt und enthält Angaben zu Erweiterungen, die während der Entwicklung von *smart* entstanden sind. Desweiteren wurden zusammen mit *smart* einige Fehler in der Implementierung des Device Managers entdeckt und behoben. Schließlich wurde die Software auf neue Plattformen portiert und an besondere Eigenschaften einiger Bandlaufwerke angepaßt.

Für die direkte Nutzung des Device Managers eines Clients stehen inzwischen drei verschiedene Benutzerinterfaces zu Verfügung: `dmui`, `xdmui` und `dmcp`.

4.1.1 `dmui` – Device Manager User Interface

`dmui` wurde bereits in [DM] vorgestellt. Dieses Benutzerinterface bietet direkten Zugang zu allen Funktionen des Device Managers. `dmui` ist ein zeilenorientiertes Hilfsmittel, das zum häufigen Gebrauch ungeeignet erscheint. Die Eingabe von Kommandos und Objektnamen ist sehr aufwendig.

4.1.2 `xdmui` – X11-based Device Manager User Interface

Benutzerinterface `xdmui` ist Nachfolger von `dmui` und bietet ebenfalls einen direkten Zugriff auf Funktionen des Device Managers. Hier lassen sich jedoch über Formulare und Auswahlménus einzelne Kommandos komfortabel und schnell zusammenstellen.

`xdmui` wurde zusammen mit *smart* entwickelt. Anhang A.2 enthält weitergehende Informationen zu Erscheinungsbild und Bedienung.

4.1.3 `dmcp` – Device Manager Copy

`dmcp` ist ein Programm zur Verwendung auf einer Shell-Kommandozeile. `dmcp` wurde zusammen mit *smart* entwickelt und ist deshalb in [DM] nicht

enthalten. Es dient zum einfachen Kopieren von Daten, die als *Resource*- oder *Alias*-Objekte vorliegen².

Eine nähere Erläuterung dieses Programms setzt Kenntnisse von Objekten und Kommandos des Device Managers voraus. Weitere Details wurden deshalb der Beschreibung des Device Managers im Anhang A.3 beigefügt.

4.2 Volume Manager

Volume Manager ist ein ständig laufendes Programm des Servers. Er übernimmt autonom die Verwaltung und Steuerung der zentralen Datensicherung. Für den automatischen und unbeaufsichtigten Betrieb benötigt der Server Informationen über zu sichernde Clients:

- Namen (Adressen) der Clients;
- Liste der zu sichernden Daten (Filesysteme) eines Client;
- Datensicherungsmethode, -programm, -parameter;
- Wann und wie oft eine Datensicherung durchgeführt werden soll.

Außerdem benötigt der Server u. a. Informationen über. . .

- Speichermedien: welche Bandlaufwerke stehen für eine Speicherung von Sicherungskopien zur Verfügung?
- Zugriffsrechte: welche Benutzer dürfen auf gesicherte Daten zugreifen?

Diese Informationen müssen von einem Administrator gesammelt und in einer entsprechenden Datenbanktabelle eingetragen werden. Er bedient sich dazu der Benutzerschnittstellen, wie sie im Abschnitt 4.3 beschrieben werden. Die Sammlung der für den Betrieb notwendigen Informationen nennt man Konfiguration. Die Details einer Konfiguration beschreibt Kapitel 6.2.3. Es folgen nun Erläuterungen, wie Datensicherungen im einzelnen durchgeführt werden.

²Der englische Begriff "*Resource*" wird vom Device Manager als Objektbezeichnung verwendet. Der deutsche Begriff "Ressource" steht in den folgenden Abschnitten für zur Verfügung stehende Arbeitsmittel bzw. Kapazitäten der zentralen Datensicherung.

4.2.1 Verwendung eines Cache

Abbildung 4.2 zeigt die Übertragung von Daten eines Client auf ein Bandmedium während einer Datensicherung.

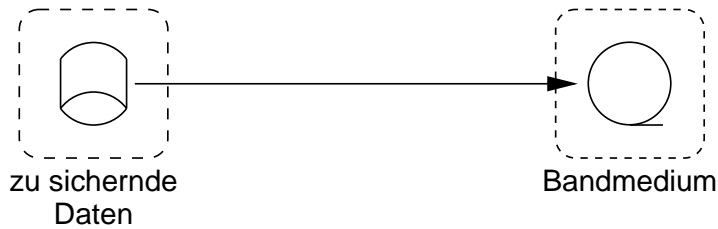


Abbildung 4.2: Datensicherung eines Filesystems auf ein Bandlaufwerk

Für den Fall, daß mehrere Datensicherungen gleichzeitig durchgeführt werden sollen, müssen hierzu eine entsprechende Anzahl von Bandlaufwerken und Medien bereitgestellt werden (Abbildung 4.3).

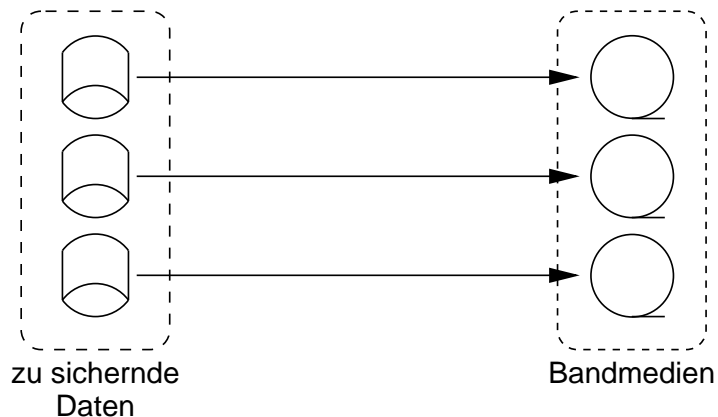


Abbildung 4.3: Datensicherung mehrerer Filesysteme auf eine entsprechende Zahl von Bandlaufwerken

Kapitel 3.6 fordert u. a. als Abhilfe für die dort gezeigten Probleme, daß die Datenströme mehrerer Clients gleichzeitig auf ein Bandlaufwerk geschrieben werden sollen.

Dies läßt sich durch sogenanntes Multiplexing erreichen (Abbildung 4.4). Datenströme der Quellen werden in kleine Teile zerlegt und dann nacheinander auf Band geschrieben. Eine Zerlegung von drei solcher Datenströme bzw. Datencontainer in kleinere Teile (Fragmente) zeigt Abbildung 4.5. Für diese Zerlegung sind Größe oder Anzahl der Fragmente unwichtig.

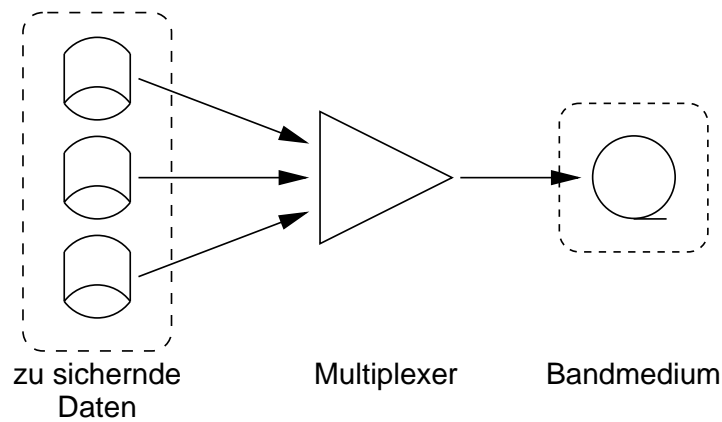


Abbildung 4.4: Datensicherung mehrerer Filesysteme auf ein Bandlaufwerk

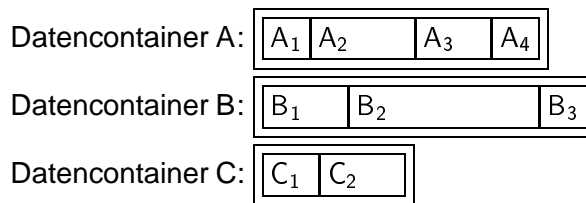


Abbildung 4.5: Aufspaltung von Datencontainern (Volumes) in kleinere Teile (Fragmente)

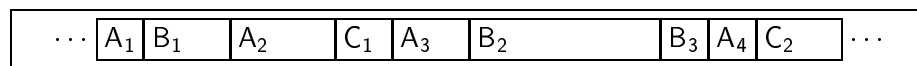


Abbildung 4.6: Mögliche Reihenfolge von Fragmenten auf einem Datensicherungsmedium

Der Volume Manager ist in der Lage, mehrere Datensicherungen gleichzeitig durchzuführen und die Fragmente der verschiedenen Volumes nacheinander auf Band zu schreiben. Die Reihenfolge in der die einzelnen Fragmente schließlich auf Band stehen werden, hängt davon ab, in welcher Reihenfolge sie von den jeweiligen Datensicherungsprogrammen bzw. Device Managern übertragen werden. Eine mögliche Reihenfolge des Bandinhalts zeigt Abbildung 4.6. Diese Reihenfolge wird von zufälligen Faktoren bestimmt. Dazu gehören die Datensicherungsgeschwindigkeit der Clients sowie Anzahl und Größe der einzelnen Fragmente. Die tatsächliche Reihenfolge von Fragmenten eines Volumes auf einem Datensicherungsmedium ist unerheblich. Wichtig ist nur ein korrekter Vermerk im Inhaltsverzeichnis des Volume Managers.

Für die Funktion des Multiplexers benötigt der Volume Manager einen Zwischenspeicher, auf dem er einzelne Fragmente vorübergehend ablegen

kann. Dieser Zwischenspeicher wird Cache genannt. Ein Cache ist in der Regel ein für den Volume Manager reservierter Speicherplatz auf einer Festplatte (Filesystem). Den Zugriff auf einen Cache wickelt der Volume Manager über die Funktionen eines Device Managers ab. Der Volume Manager füllt diesen Cache mit Daten wie folgt:

1. Volume Manager öffnet eine neue Datei im Cache als *File-Resource* über den zuständigen Device Manager³.
2. Er startet über den Device Manager eines Clients die gewünschte Datensicherung. Als Ziel für die Übertragung des Datenstroms gibt der Volume Manager die neue Datei an und legt eine maximale Größe für diese Datei fest⁴.
3. Die beteiligten Device Manager (Quelle und Ziel) beginnen mit der Datenübertragung.
4. Erreicht die Cache-Datei die angegebene Größe bricht der Datentransport ab. Der Volume Manager öffnet nun wie in Schritt 1 eine neue Datei im Cache und läßt den unterbrochenen Datentransport mit diesem neuen Ziel fortfahren.

Sobald das Schreiben einer Cache-Datei beendet ist (maximale Größe erreicht oder Datensicherung beendet), beginnt der Volume Manager diese auf ein Medium in einem Bandlaufwerk zu kopieren. Nach Abschluß des Kopiervorgangs löscht er die Cache-Datei und gibt den durch sie belegten Speicherplatz für weitere neue Cache-Dateien frei. Das Kopieren einer Cache-Datei auf Band erfolgt unabhängig vom Füllen anderer Cache-Dateien.

Es gibt Situationen in denen mehrere Cache-Dateien zum Kopieren auf Band bereitstehen. Der Volume Manager kopiert in diesen Fällen alle Cache-Dateien der Reihe nach auf Band und kann dabei die Maximalgeschwindigkeit des Bandlaufwerks ausnutzen.

Durch die Verwendung von Cache-Dateien zum Multiplexen paralleler Datenströme lassen sich mehrere Datencontainer quasi gleichzeitig auf Band schreiben. Der Volume Manager entscheidet dabei selbstständig wieviele Datensicherungen er starten darf, damit die zur Verfügung stehenden Ressourcen optimal ausgenutzt werden. Diese Entscheidung trifft er aufgrund der Anzahl zur Verfügung stehender Ressourcen (Cache-Dateien, Bandlaufwerke).

³Der jeweils für eine *Resource* zuständige Device Manager befindet sich auf demselben Rechner, zu dem auch die *Resource* gehört.

⁴Eine Größenbeschränkung teilt der Volume Manager dem Device Manager über die `limit`-Option des `transport`-Kommandos mit.

4.2.2 Fragmente

Fragmente wurden bereits im Zusammenhang mit der Verwendung eines Cache vorgestellt. Eine Teilung von Datencontainern in Fragmente hat noch einen weiteren Vorteil. Nur auf diese Weise ist es möglich, Datencontainer auf Bandmedien zu sichern, die größer sind als die Kapazität eines einzelnen Mediums.

Beispiel: Es soll ein Filesystem einer großen Festplatte mit vier GByte Kapazität in die zentrale Datensicherung einbezogen werden. Dort stehen jedoch nur Bandgeräte zur Verfügung, die maximal ein GByte auf einem Medium speichern können. In diesem Fall muß der Datencontainer einer Gesamtsicherung⁵ in mindestens vier Fragmenten à ein GByte geteilt werden.

Eine "Zersplitterung" von Datencontainern in viele kleine Fragmente (Abbildungen 4.5 und 4.6) und deren mögliche "Zerstreuung" über mehrere Medien (obiges Beispiel) scheinen Nachteile des Volume Managers zu sein. Eine Restaurierung dieser zerstreuten Fragmente ist schwieriger und zeitaufwendiger im Vergleich mit einer Restaurierung einer kompletten, unfragmentierten Sicherungskopie. Mit Hilfe einer geeigneten Benutzerschnittstelle läßt sich jedoch dieser Nachteil ausgleichen. Wie eine Restaurierung mit Hilfe der *smart*-Benutzerschnittstellen durchgeführt wird, stellt Kapitel 5 vor.

4.2.3 Sicherungsintervall

Wie bereits erwähnt, eignet sich für eine zentrale Datensicherung die in Abschnitt 3.2 vorgestellte Datensicherungsmethode 9. Dabei muß diese Methode für jeden Client bzw. Filesystem regelmäßig angewendet werden. Der Begriff "regelmäßig" soll nun genauer beschrieben werden.

Eine regelmäßige Sicherung von Daten bedeutet, daß die zur Sicherung von Daten ausgewählte Methode in festgelegten Zeitabständen angewendet wird. Üblicherweise werden dabei wichtige Daten täglich oder sogar zwei- oder mehrmals täglich gesichert. Für unbedeutendere Datenbestände genügt meist ein wöchentlicher Turnus.

Die Zeitdauer zwischen zwei aufeinanderfolgenden Datensicherungen wird unter *smart* als Intervall bezeichnet. Abschnitt 4.2.5 erklärt, daß es sich bei diesem Intervall um eine Zeitangabe für die Planung einer Datensicherung handelt, der von der tatsächlichen Zeitspanne zwischen zwei Sicherungen abweichen kann.

⁵Annahme: das Filesystem ist zu 100 % mit Daten gefüllt.

4.2.4 Zeitrahmen

Eine regelmäßige Datensicherung wird zum einen von einem Intervall bestimmt. Angabe wie z. B. "täglich" genügen jedoch zur Durchführung von Sicherung nicht. Als zusätzliche Information benötigt der Volume Manager eine Zeitangabe (z. B. Tageszeit), zu der eine Sicherung gestartet und ab der mit Hilfe des angegebenen Intervalls der Zeitpunkt der nächsten Sicherung berechnet werden soll.

Eine Datensicherung belegt wertvolle Ressourcen: Netzwerkbandbreite während der Übertragung von Datencontainern, CPU-Zeit auf den Clients für die Datensicherungsprogramme, Bandlaufwerk des Servers, etc. Es ist daher sinnvoll, eine Datensicherung zu einer Zeit durchzuführen, zu der auf dem Netzwerk und den Clients "wenig los ist", z. B. während der Nachtstunden oder am Wochenende. Mit einer geschickten Wahl der Sicherungszeiten lassen sich die zur Verfügung stehenden Ressourcen besser auslasten.

Ein Intervall und Zeitpunkt, wie z. B. täglich um 22 Uhr, genügen als Angaben zur Festlegung einer einzelnen Sicherung. Jedoch muß eine zentrale Datensicherung eine Vielzahl solcher Sicherungen bearbeiten können. Eine größere Zahl von Datensicherungen wie z. B. die der Fakultät Informatik mit 86 Clients und über 460 Filesystemen⁶ können nicht zu einem festen Zeitpunkt gesichert werden. Dazu reichen die Kapazitäten des Netzwerks und des Servers der zentralen Datensicherung nicht aus.

Statt eines Zeitpunkts eignet sich daher die Angabe eines Zeitraumes, über den sich die durchzuführenden Sicherungen verteilen lassen: z. B. nächtliche Sicherung aller angemeldeter Filesysteme/Clients in der Zeit von 20 Uhr bis 6 Uhr am nächsten Morgen. Damit bleibt es dem Volume Manager überlassen, wann er die Datensicherung eines einzelnen Filesystems startet.

Die Angabe eines solchen Zeitraums wird unter *smart* als Zeitrahmen bezeichnet. Ein großer Vorteil von Zeitrahmen ist die dadurch mögliche Flexibilität. Der Volume Manager kann auf Störungen des Netzwerks oder einzelner Clients reagieren, indem er deren Datensicherung kurzzeitig verschiebt. Er wird versuchen eine mißglückte Sicherung eines Filesystems/Clients an einem späteren Zeitpunkt innerhalb des angegebenen Zeitraums zu wiederholen. Er ist auch in der Lage mehrere durchzuführende Datensicherungen nach verschiedenen Kriterien zu sortieren und dann nacheinander abzuarbeiten. Kriterien für eine Sortierung werden in einem späteren Abschnitt vorgestellt.

Zeitrahmen beziehen sich grundsätzlich nur auf den Startzeitpunkt einer Datensicherung. Der Volume Manager kontrolliert nicht deren Dauer. Es

⁶Stand Anfang November 1996

ist daher durchaus möglich, daß eine Datensicherung kurz vor Ende eines Zeitrahmens vom Volume Manager gestartet wird und weit über diesen Zeitraum hinaus andauert.

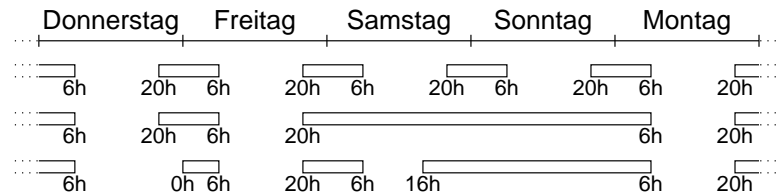


Abbildung 4.7: Zeitpunkt einer Datensicherung festgelegt durch Zeitrahmen

Abbildung 4.7 enthält Beispiele solcher Zeitrahmen. Ein einzelner Zeitrahmen täglich wiederholt wie z. B. 20 bis 6 Uhr (erste Zeile) genügt nicht für jeden Anwendungsfall. *smart* bietet deshalb die Möglichkeit für jedes zu sichernde Filesystem verschiedene Zeitrahmen zu wählen. Auf diese Weise können z. B. für jeden Wochentag einzelne Zeitrahmen den täglichen Arbeitszeiten der Mitarbeiter eines Unternehmens angepaßt werden.

Ein Zeitrahmen darf mehrere Tage umfassen, z. B. ein Wochenende von Freitag Abend bis Montag Morgen (mittlere Zeile). Damit läßt sich eine wöchentliche durchzuführende Sicherung einfach auf das Wochenende festlegen.

Die in Abbildung 4.7 unten dargestellten Zeitrahmen enthalten individuelle Angaben für einzelne Wochentage: Donnerstag erst ab Mitternacht sowie Samstag bereits ab 16 Uhr.

4.2.5 Kombination von Zeitrahmen und Intervall

Wie im letzten Abschnitt 4.2.4 beschrieben wurde, berechnet sich nach einer Datensicherung der Zeitpunkt für die nächste Sicherung aus Intervallangabe und vorgegebenem Zeitrahmen. Diese Berechnung erscheint einfach: Zum Zeitpunkt des Starts der letzten Sicherung addiert man die Intervalllänge und erhält den neuen Zeitpunkt.

Durch das dynamische Verhalten des Volume Managers (kurzfristiges Verschiebung von Datensicherungen innerhalb eines Zeitrahmens) können einzelne Sicherung ausfallen! Als Beispiel dafür zeigt Abbildung 4.8 die Anwendung eines täglichen Intervalls (Intervalldauer 24h) bei einem Zeitrahmen von 20 Uhr bis 6 Uhr an Werktagen und einem vergrößerten Zeitrahmen von 20 Uhr bis 9 Uhr an Wochenenden.

Den Zeitpunkt einer Datensicherung markieren in der Abbildung senkrechte Pfeile. Demnach wurde die erste Sicherung vom Volume Manager am

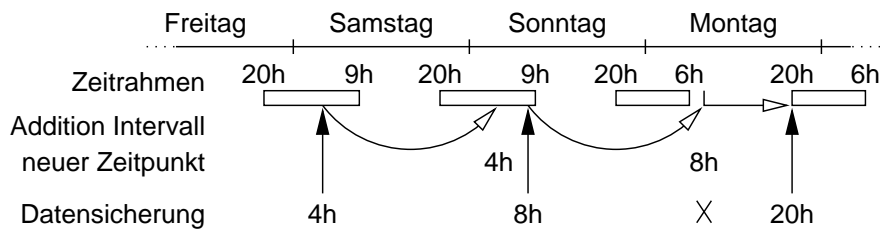


Abbildung 4.8: Berechnung eines Sicherungszeitpunkts aus Zeitpunkt der letzten Sicherung, Zeitrahmen und Intervall

Samstag gegen 4 Uhr durchgeführt. Mit einem Intervall von 24h wird ein neuer Zeitpunkt Sonntag, 4 Uhr berechnet. Aufgrund verschiedener Faktoren kann sich der Start der nun folgenden Datensicherung noch etwas verzögern. Mögliche Gründe für Verzögerungen wurden bereits aufgezählt. Es handelt sich dabei z. T. um Betriebsstörungen, auf die der Volume Manager keinen Einfluß hat. Weitere Gründe nennt Abschnitt 4.2.6. In unserem Beispiel startet der Volume Manager am Sonntag gegen 8 Uhr die zweite Datensicherung rechtzeitig vor Ende des gerade gültigen Zeitrahmens.

Als Startzeit für die dritte Datensicherung wird aus dem Zeitpunkt der zweiten Sicherung berechnet: Montag, 8 Uhr. Allerdings liegt dieser Termin außerhalb des erlaubten Zeitrahmens. Der Volume Manager muß daher die Datensicherung bis zum Beginn des nächsten Zeitrahmens am Montag Abend verschieben. Damit ist eine gewünschte Datensicherung der Nacht von Sonntag auf Montag ausgefallen.

Zur Vermeidung eines solchen Ausfalls von Datensicherungen werden Intervalle immer auf den Beginn eines Zeitrahmens bezogen. Damit erhalten wir aus dem zuvor angeführten Beispiel neue Zeitangaben wie in Abbildung 4.9 dargestellt.

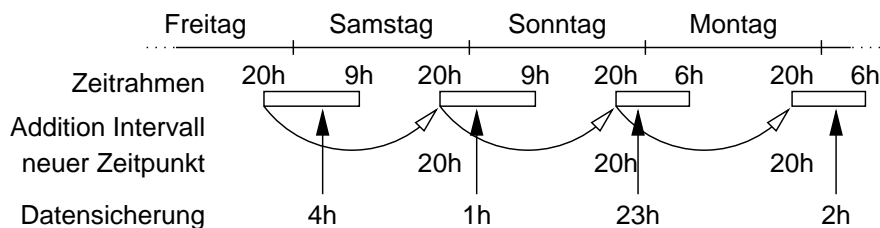


Abbildung 4.9: Berechnung eines Sicherungszeitpunkts aus Anfangszeit eines Zeitrahmens und Intervall

Durch die Berechnung der Zeitpunkte aus einem Intervall, bezogen auf den Start eines Zeitrahmens, kann der Volume Manager, wie in Abbildung 4.9

gezeigt, die vorgegebenen Zeitrahmen voll ausnutzen und alle Sicherungen wie gewünscht täglich durchführen.

Intervall und Zeitrahmen lassen sich beliebig miteinander kombinieren. Dabei sind einige Sonderfälle zu beachten. Abbildung 4.10 demonstriert einen solchen Sonderfall: Innerhalb eines Zeitrahmens von täglich 14 Uhr bis 10 Uhr am nächsten Tag wird ein Intervall von 7 Stunden mehrmals wiederholt. Die Berechnung der Sicherungszeitpunkte erfolgt wie zuvor bezogen auf den Beginn eines Zeitrahmens. Ist ein Intervall kürzer als der Zeitrahmen selbst, wird das Intervall entsprechend wiederholt.

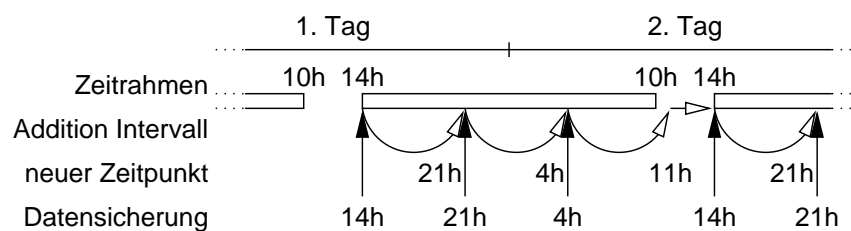


Abbildung 4.10: Berechnung eines Sicherungszeitpunkts; mehrere Intervalle innerhalb eines Zeitrahmens

Für den Fall, daß sich im dargestellten Beispiel eine Datensicherung um einige Stunden verzögert, besteht die Möglichkeit, daß die nachfolgende Datensicherungen unmittelbar danach durchgeführt wird.

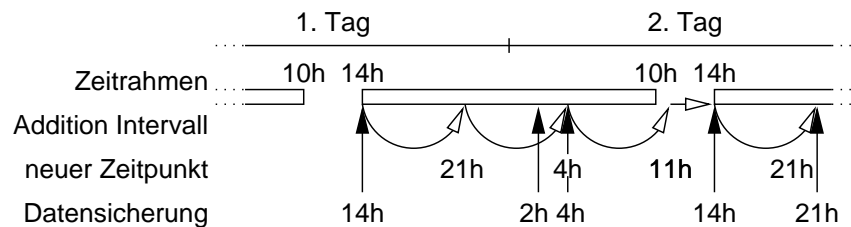


Abbildung 4.11: Berechnung eines Sicherungszeitpunkts; Zeit zwischen zwei Sicherungen kann zu kurz werden

Abbildung 4.11 zeigt einen solchen Fall. Im Vergleich mit Abbildung 4.10 verzögert sich hier die zweite Datensicherung um 5 Stunden bis gegen 2 Uhr. Der Zeitpunkt der dritten Sicherung ist 4 Uhr. Der Volume Manager berechnet diesen aus Anfangszeit des Zeitrahmens (14 Uhr) und Addition zweier Intervalle (à 7 Stunden). Damit verbleiben zwischen zweiter und dritter Sicherung nur 2 Stunden Zeit.

Schnell aufeinanderfolgende Sicherungen derselben Daten, wie im gezeigten Beispiel, sind unerwünscht. Deshalb wartet der Volume Manager nach

4.2.6 Priorität

Es kommt häufig vor, daß der Volume Manager zu einem Zeitpunkt viele Datensicherungen gleichzeitig durchführen soll. Sobald deren Anzahl die zur Verfügung stehenden Ressourcen übersteigt, müssen ein Teil der geplanten Sicherungen kurzzeitig verschoben werden, bis wieder genügend Ressourcen freigeworden sind. Dies stellt einen weiteren Grund für Verzögerungen von Datensicherungen dar.

Die Entscheidung, ob eine Sicherungen verschoben oder sofort durchgeführt wird, trifft der Volume Manager mit Hilfe von Prioritäten. Jeder geplanten Datensicherung wird eine solche Priorität zugeordnet.

Bei der Konfiguration einer Datensicherung wird eine Priorität vom Operator angegeben. Sinnvollerweise wählt man für eine Sicherung wichtiger Daten hohe Zahlen, für unwichtige dagegen kleine Zahlen als Priorität. Zu dieser Prioritätsangabe (Anfangspriorität) wird noch ein "Verzögerungsfaktor" addiert. Sein Wert ist der Quotient aus der bisherigen Verzögerung einer Datensicherung und der Länge des Intervalls. Die Verzögerung ist die Zeitdifferenz aus geplantem Start einer Sicherung und der aktuellen Zeit. Formal ergibt sich für die Priorität einer Datensicherung:

$$\text{Priorität} = \text{Anfangspriorität} + \frac{\text{Verzögerung}}{\text{Intervall}}$$

Für eine Datensicherung, die bereits für die Zeitdauer eines Intervalls verzögert wurde, erhöht sich die, in der Planung berücksichtigte Priorität, um 1. Bei Sicherungen, die bisher noch nicht verzögert wurden, beträgt der "Verzögerungsfaktor" 0 (Null). Jede Verzögerung wirkt sich also mit einem entsprechenden Anteil auf die Priorität aus.

Die Liste, aller für einen Zeitpunkt eingeplanten Datensicherungen, wird vom Volume Manager nach diesen Prioritäten sortiert. Die sortierte Liste arbeitet er der Reihe nach ab, beginnend bei einer Datensicherung mit höchster Priorität. Dabei wird die Liste ständig aktualisiert und die Prioritäten neu berechnet.

Eine Ausnahme stellt die Anfangspriorität mit dem Wert Null dar. Eine Datensicherung mit dieser Priorität wird vom Volume Manager nicht berücksichtigt. Dieser Wert kann dazu verwendet werden, einen Eintrag zur Datensicherung in der Datenbank vorübergehend "außer Betrieb" zu setzen, ohne den Eintrag ganz löschen zu müssen. Einen anderen Einsatzzweck zeigt der nächste Abschnitt 4.2.7.

Bei der Entscheidung, ob eine Sicherung sofort durchgeführt werden kann, oder ob sie zu verschieben ist, berücksichtigt der Volume Manager die Vorgabe, daß für jeden Client immer nur eine Datensicherung gestartet werden

darf. Daraus folgt, daß mehrere Filesysteme eines Clients nie gleichzeitig gesichert werden.

Eine Priorität ist ebenfalls bei der Konfiguration von Cache-Dateien und Bandlaufwerken verwendbar. In diesem Falls wird ihre Nutzung beschränkt auf Datensicherungen, welche mindestens die angegebene Priorität besitzen. Es lassen sich so, bei geschickter Vergabe von Prioritäten, Cache und Bandlaufwerke als Ressourcen für einen Teil des zu sichernden Datenbestands reservieren.

4.2.7 Verbinden mehrerer Datensicherungen

Inhaltlich zusammengehörende Daten werden häufig auf mehreren Filesystemen verteilt gespeichert. Das hat zur Folge, daß mit einigen Datensicherungsprogrammen (z. B. `dump`) diese Daten nur nach Filesystemen getrennt gesichert werden können. Dagegen ist es wünschenswert, diese Datenbestände gemeinsam zu sichern. Zumindest sollte die Zeit zwischen den Sicherungen der einzelnen Filesysteme möglichst kurz gehalten werden.

Der Volume Manager bietet deshalb eine Möglichkeit, Einträge dieser Filesysteme miteinander zu verbinden. Bei der Konfiguration kann für jeden Eintrag ein weiterer Eintrag als Verweis (Verbindung) angegeben werden. Drei und mehr Filesysteme können unter Bildung einer Kette von Verbindungen zusammengefaßt werden (Beispiel: Filesystem A ist verbunden mit B, B ist verbunden mit C usw.).

Sobald der Volume Manager eine Datensicherung startet, die eine solche Verbindung aufweist, setzt er die Priorität, der mit ihr verbunden anderen (zweiten) Datensicherung, auf einen Maximalwert. Die im letzten Abschnitt 4.2.6 beschriebene Liste wird daraufhin mit der geänderten Priorität neu sortiert. Dadurch wird erreicht, daß diese zweite Sicherung aufgrund ihrer Priorität als nächstes gestartet wird, sofern entsprechende Ressourcen frei sind.

Im letzten Abschnitt wurden ebenfalls Einträge zur Datensicherung in der Datenbank mit Anfangspriorität Null beschrieben. Solche Einträge sind passiv, d. h. sie werden vom Volume Manager ignoriert. Wenn jedoch ein solcher Eintrag mit einer anderen Datensicherung verbunden ist und diese andere Datensicherung gerade gestartet wurde, dann wird der bisher passive Sicherungseintrag mit höchster Priorität aktiviert.

Mit dieser Funktionalität können Filesysteme zur Datensicherung konfiguriert werden, die nur dann gesichert werden, wenn ein anderes mit ihm verbundenes Filesystem gerade gesichert wurde.

Voraussetzung für Datensicherungen im Verbund sind geeignete Zeitrahmen und Intervalle für jedes verbundene Filesystem, die eine Datensicherung zu diesem Zeitpunkt erlauben.

4.2.8 Gruppen

Zur Festlegung von Zugriffsrechten einzelner Administratoren werden Gruppen verwendet. Jeder Administrator ist einer oder mehreren Gruppen zugeordnet. Ebenso wird mit jedem Eintrag zur Datensicherung eines Clients eine oder mehrere Gruppen verknüpft.

Eine Liste von Gruppen ist eine sogenannte Access-Control-List (ACL). Eine Gruppenbezeichnung besteht aus einem einfachen Namen. Ein besonderer Gruppenname ist "Operator". Einige Funktionen der zentralen Datensicherung sind Administratoren vorbehalten, die der Gruppe "Operator" angehören. Zu diesen Funktionen gehören u. a. Konfiguration und Verwaltung der Bandmedien (z. B. Wechsel von Medien der Bandlaufwerke).

Bei Einsatz von *smart* zur zentralen Datensicherung mehrerer Abteilungen empfiehlt es sich, für jede Abteilung eine eigene Gruppe einzurichten. *smart* bietet für eine Betriebsstatistik und Kostenstellenrechnung u. a. eine nach Gruppen getrennte Auswertung an. Administratoren einer Abteilung haben, bei entsprechender Konfiguration, Zugang zu Protokollen und Informationen über die aus ihrer Abteilung gesicherten Daten. Zudem können sie über eine Benutzerschnittstelle eine Restaurierung von Daten der Abteilung anfordern.

Cache-Dateien und Bandlaufwerke lassen sich ebenfalls einer Liste von Gruppen zuordnen. Analog zur Verwendung von Prioritäten wird die Nutzung eines Cache oder Bandlaufwerks beschränkt auf Datensicherungen, welche mindestens einer der aufgezählten Gruppen angehören. Sinnvoll können dadurch z. B. große Filesysteme einzelner Abteilungen als Cache für die zentrale Sicherung des Datenbestands dieser Abteilung eingesetzt werden.

4.2.9 Aufbewahrungsfristen

Jeder Datensicherung wird eine, bei der Konfiguration festgelegte, Archivierungsfrist (Aufbewahrungsfrist) zugeordnet. Diese Archivierungsfrist ist ein beliebiger Text, der eine Klassifizierung der gesicherten Daten erlaubt. Es handelt sich also nicht um die Angabe eines Datums oder einer Zeitspanne.

Übliche Texte zur Verwendung als Aufbewahrungsfrist sind z. B. "Standard", "Kurzfristig". Eine Angabe von "2 Jahre" hat dabei keinerlei direkten Einfluß auf die Zeitdauer der Archivierung einzelner Medien.

Die Angabe einer Aufbewahrungsfrist wird von *smart*-Hilfsprogrammen genutzt. Ein Operator kann mit diesen Programmen das Medienarchiv (genauer: die Inhaltsverzeichnisse der Datenbank) durchsuchen und nicht mehr benötigte Sicherungen zur Löschung vormerken. Diese Löschung von gesicherten Daten besteht dabei lediglich aus dem Entfernen des entsprechenden Eintrags aus dem Inhaltsverzeichnis der Datenbank.

Wenn alle auf einem Medium abgelegten Daten aus dem Inhaltsverzeichnis der Datenbank gelöscht wurden, teilt *smart* einem Operator mit, daß dieses Medium keine Daten mehr enthält, die aufbewahrt werden müssen. Der Operator kann daraufhin das entsprechende Medium aus dem Bandarchiv entfernen.

Aufbewahrungsfristen werden z. T. durch gesetzliche Regelungen bestimmt. So müssen bestimmte Datensicherungsbestände mindestens für eine festgelegte Zeit aufbewahrt werden, wogegen andere Datenbestände innerhalb einer Löschfrist aus dem Archiv zu entfernen sind⁸.

4.2.10 Datensicherungsprogramme

Datensicherungsprogramme auf den Clients sind zuständig für die "Verpackung" der zu sichernden Daten als Container. Einige dieser Programme orientieren sich bei der Sicherung an Filesystemen der Clients. Alle Daten eines Filesystems werden dabei gemeinsam gesichert. Das gebräuchlichste Sicherungsprogramm ist das vom Betriebssystem BSD stammende `dump`-Kommando, das unter den meisten modernen UNIX-Betriebssystemen zur Verfügung steht. Die Implementationen und damit das Verhalten des Sicherungsprogramms ist z. T. auf den einzelnen Betriebssystemen sehr unterschiedlich.

Neben `dump` gibt es noch andere Sicherungsprogramme wie z. B. `tar`, `cpio`, `fbackup`, sowie Variationen dieser Programme als `gtar`, `afio` oder `backup`.

Alle aufgezählten Sicherungsprogramme können nicht direkt zur Datensicherung nach der in Kapitel 3.2 beschriebenen Sicherungsmethode 9 eingesetzt werden. Einige Programme benötigen zur Datensicherung die Angabe eines `dump levels` (z. B. `dump`, `gtar`). Zum Teil verlangen Sicherungsprogramme die Angabe des zu sichernde Datenbestand, welche dem Programm als Liste anzugeben ist. Diese Nachteile lassen sich umgehen, indem Sicherungsprogramme nicht direkt von *smart* benutzt werden. Statt-

⁸[IT 96, Seite M6-29] enthält eine Aufstellung von Gesetzen über Archivierungsfristen.

dessen finden in der Konfiguration von Datensicherungen entsprechende Skripte Verwendung, welche die für Methode 9 notwendigen Vorarbeiten erledigen können. Diese Vorarbeiten bestehen aus...

- Durchsuchen des zu sichernden Datenbestands (Filesystem) auf Änderungen seit der letzten Sicherung,
- Festlegung des nächsten dump levels aufgrund vorgegebener Parameter und des Suchergebnisses sowie
- Aufruf eines Sicherungsprogramms mit den erforderlichen Optionen und Argumenten.

Für diese Vorarbeiten benötigen die Skripte Informationen, die vom Volume Manager als Kommandoargumente übergeben werden. Dazu gehören u. a. Informationen über zu sichernde Daten, gewünschte Sicherungssequenz und Schwellwert.

Neben der Vorbereitung einer Datensicherung haben diese Skripte noch eine andere Aufgabe. Sie analysieren das Protokoll des Sicherungsprogramms auf Fehlermeldungen oder Störungen während der Datensicherung. Folglich prüfen sie auf einen korrekten Programmablauf. Der Volume Manager trägt alle Meldungen dieser Skripte zusammen mit Informationen über die Datensicherung in das Inhaltsverzeichnis der Datenbank ein.

Es wurde bereits beschrieben, wie der Volume Manager den Startzeitpunkt einer Datensicherung innerhalb eines Zeitrahmens festlegt. Eine Sicherung darf dabei weit über das Ende eines Zeitrahmens hinaus andauern, ohne daß der Volume Manager das Sicherungsskript abbricht. Falls jedoch eine Sicherung aufgrund von Störungen oder Programmfehlern an einer Stelle der Datensicherung "hängen bleibt", greift der Volume Manager nach Ablauf eines Timeouts (Wartezeit) ein und bricht die Datensicherung ab.

Zur Führung von Inhaltsverzeichnissen gesicherter Daten benötigt der Volume Manager einige Informationen über jede durchgeführte Sicherung. Dazu muß ein Datensicherungsskript diese Informationen in einer vorgeschriebenen Form als Protokollausgabe (Ausgabekanal stderr) melden. Die Meldung jeder Information erfolgt jeweils als vollständige Textzeile. Damit der Volume Manager die gesuchten Informationen zum Eintrag in die Datenbank leicht aus einem längeren Protokoll herausfinden kann, ist das Format dieser Textzeilen vorgeschrieben:

```
VM-Info-<item>=<value>
```

Anstelle von *<item>* und *<value>* muß das Sicherungsskript entsprechende Angaben einsetzen:

- “VM-Info-LEVEL=...”: einstellige Zahl, der vom Sicherungsskript gewählte dump level; Beispiel für eine Gesamtsicherung:

```
VM-Info-LEVEL=0
```

- “VM-Info-ID=...”: Ein vom Sicherungsskript gewählter Identifier für den erzeugten Datencontainer (Dump-ID oder “timestamp”). Das Skript kann z. B. die vom Programm `dump` ausgegebene Zeichenkette für “dump date” verwenden. Anhang A.1.1 beschreibt, welche Zeichen in einem Identifier vorkommen dürfen. Dieser Identifier darf vom der vergebenen Volume-ID verschieden sein und wird nur vom Sicherungs- bzw. Restaurierungsprogramm zu Verwaltungszwecken genutzt. Zwischen Volume-ID (vom Volume Manager gewählter Identifier) und Dump-ID (vom Sicherungsskript gewählter Identifier) ist eine eindeutige Zuordnung möglich⁹.
- “VM-Info-REQUIRE=...”: Eine inkrementelle Sicherung eines bestimmten dump level setzt die Existenz weiterer Datensicherungen eines kleineren dump levels voraus. Die REQUIRE-Angabe bezeichnet den Identifier der Sicherung, die als Basis (Voraussetzung) für die aktuelle Sicherung dient. Beispiel: Das Programm `dump` gibt bei einer inkrementellen Datensicherung u. a. folgenden Text aus:

```
DUMP: Date of this level 3 dump: Fri Nov 22 22:42:55 1996
DUMP: Date of last level 0 dump: Mon Nov 18 22:37:57 1996
```

Daraus könnte das Sicherungsskript folgende Meldungen formen und an den Volume Manager schicken:

```
VM-Info-LEVEL=3
VM-Info-ID=      19961122.224255
VM-Info-REQUIRE= 19961118.223757
```

Leerzeichen nach dem “=”-Zeichen werden vom Volume Manager ignoriert. REQUIRE-Angaben vergleicht er mit den ID-Angaben früherer Sicherungsläufe. Falls der angegebene Identifier unbekannt ist, gibt der Volume Manager eine Fehlermeldung aus.

- “VM-Info-STATUS=...”: Der Volume Manager benötigt eine Statusinformation, ob das Datensicherungsprogramm den Datencontainer ohne Fehler erzeugen konnte. In diesem Fall erwartet er nach dem Gleichheitszeichen das Wort “ok”. Jeder andere Text an dieser Stelle wird als Fehlermeldung interpretiert.

⁹Der Grund für eine Verwendung einer vom Sicherungsprogramm abhängigen Dump-ID beruht auf einer dadurch möglichen Vereinfachung der Sicherungsskripte.

- “VM-Info-TEXT=...”: Diese Zeilen dürfen einen beliebigen Text enthalten, der vom Volume Manager mit in das Inhaltsverzeichnis aufgenommen wird. Für die Funktion der Datensicherung hat dieser Text keine Bedeutung und dient lediglich zur Information der Administratoren. Diese Textzeilen sollten nur kurze Informationen (z. B. Zusammenfassung des Sicherungsprotokolls) über die Datensicherung enthalten, da sonst der Speicherplatzbedarf in der Datenbank zu groß werden kann. Bei Auftreten von Fehlern während einer Datensicherung, können über diese Textzeilen ausführliche Protokolle gemeldet werden.
- “VM-Info-SYSTEM=...”: Angaben zum Betriebssystem des Client und dessen Version. Diese Information kann bei einer Restaurierung von Daten hilfreich sein.
- “VM-Info-TYP=...”: Angaben zum Typ des Datencontainers bzw. des Sicherungsprogramms, z. B.: “bsd-dump”, “hp-dump”, “hp-fbackup” oder “gnu-tar”. Mit Hilfe dieser Angabe läßt sich eine teilweise automatisierte Restaurierung von Daten realisieren.

Die VM-Info-Angaben ID, REQUIRE, TEXT, SYSTEM und TYP ist optional, d. h. sie dürfen entfallen. Die Angabe von STATUS und LEVEL ist dagegen zwingend vorgeschrieben. Fehlen diese Angaben, geht der Volume Manager davon aus, daß die Datensicherung fehlerhaft oder unvollständig war. Fehlermeldungen gibt der Volume Manager über eine Console (siehe Abschnitt 3.5) an einen Operator weiter.

Alle Meldungen der Sicherungsskripte, die nicht das beschriebene Format “VM-Info- . . .” haben, werden normalerweise vom Volume Manager ignoriert. Nur bei Auftreten von Fehlern und wenn keine “VM-Info-TEXT”-Zeilen im Protokoll enthalten sind, trägt der Volume Manager das gesamte Protokoll in das Inhaltsverzeichnis der Datenbank ein.

Es stehen portabel geschriebene Shell-Skripts zur Verfügung, welche die genannten Datensicherungskommandos `dump` oder `gtar` sowie deren Varianten anwenden können und die vom Volume Manager geforderten Meldungen generieren. Nähere Informationen darüber ist dem *smart*-Softwarepaket zu entnehmen.

Bisher wurden nur Programme und Skripte zur Datensicherung behandelt. Eine zentrale Datensicherung benötigt selbstverständlich auch Restaurierungsprogramme. Das *smart*-Softwarepaket enthält zu jedem verwendeten Sicherungsprogramm und -Skript ein dazu passendes Restaurierungsprogramm. Datensicherungsprogramme sind auf automatischen Betrieb ausgelegt. Dagegen werden die Restaurierungsprogramme über eine Benutzerschnittstelle interaktiv von einem Operator gesteuert. Restaurierung von Daten beschreibt Kapitel 5.

4.3 *smart*-Benutzerschnittstelle

Unter *smart* stehen mehrere Benutzerschnittstellen zur Verfügung. Abbildung 4.14 zeigt wie ein Administrator mit Hilfe der bisher vorgestellten Programme *dmui*, *xdmui* und *dmcp* auf Funktionen der Device Manager (DM) zugreifen kann.

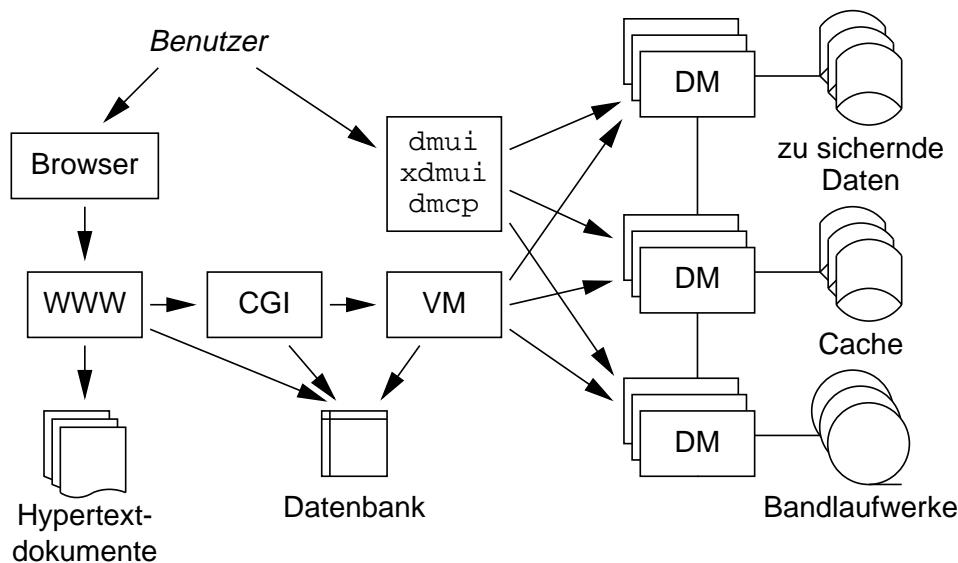


Abbildung 4.14: Zentrale Datensicherung mit *smart* – beteiligte Instanzen

Ein Volume Manager (VM) gibt ebenfalls Kommandos an die Device Manager und steuert damit die Datensicherungen. Sein Vorgehen wird von Informationen bestimmt, die in der Datenbank zu finden sind. Diese Informationen kann ein Administrator über ein weiteres Benutzerinterface in die Datenbank eintragen. Außerdem kann ein Administrator darüber Funktionen des Volume Managers in Anspruch nehmen.

Für dieses Benutzerinterface wird ein WWW-Server (World Wide Web) zu Hilfe genommen. WWW-Server erlauben einen netzweiten Zugriff auf ein lokales Informationssystem. Ein Benutzer verwendet, zum Zugriff auf über WWW angebotene Informationen, einen sogenannten Browser¹⁰. Informationen eines WWW-Servers sind als Hypertextdokumente gestaltet. Aus diesem Grund sind alle Funktionen der *smart*-Benutzerschnittstelle über Hypertextdokumente erreichbar. Ein Abrufen von Hypertextdokumenten läuft wie folgt ab:

¹⁰Gebräuchliche Browser im UNIX-Bereich sind Netscape, Mosaic und Lynx.

- Ein Administrator (Benutzer) wählt mit Hilfe des Browsers die gewünschten Informationen aus.
- Der Browser fordert den WWW-Server auf, das gewünschte Dokument an den Browser zu übertragen.
- Der Benutzer kann den Inhalt des Dokuments lesen, sobald es der WWW-Server übertragen hat und der Browser es auf einem Bildschirm darstellt.

Viele Hypertextdokumente haben einen statischen Charakter, d. h. ihr Inhalt verändert sich nicht. Bei jedem Abruf eines statischen Dokuments übermittelt der WWW-Server denselben Inhalt, der üblicherweise in einer Datei des WWW-Server abgelegt ist.

Daneben gibt es dynamische Dokumente, deren Inhalt vom WWW-Server auf Abruf jedesmal neu zusammengestellt werden. Eine Generierung dynamischer Dokumente erledigen sogenannte CGI-Skripte¹¹. Es handelt sich dabei um Programme, welche das gewünschte Dokument "schreiben", basierend auf den vom Benutzer/Browser übermittelten Angaben (z. B. Adressen, Filenamen, Suchbegriffe, etc.).

CGI-Skripte unter *smart* bieten folgende Funktionen an:

- Datensicherungsplanung: Neueintrag, Ändern und Löschen von Einträgen einzelner Clients und zu sichernder Datenbestände.
- Anzeige des Zustands von zur Zeit laufenden bzw. geplanten Datensicherungen; Abbrechen einer Datensicherung.
- Ausgabe der vom Volume Manager geführten Inhaltsverzeichnisse; Suche in den Verzeichnissen
- Verwaltung des Medienarchivs; Ausgabe von Inventurlisten; Drucken von Medienlabel.
- Bereitstellen gesicherter Daten zur Restaurierung.
- Einmaliges Sichern eines Datencontainers außerhalb der regulären Datensicherung.
- Kopieren gesicherter Datencontainer auf ein anderes Medium.
- Erstellen von Statistiken über die zentrale Datensicherung sowie einzelner Clients.
- Kostenstellenrechnung: Aufteilung von Betriebskosten nach Abteilung, Client oder Filesystem.

¹¹CGI steht für "Common Gateway Interface".

- Console: Meldungen des Volume Managers an einen Operator.
- Bandwechsel: An- und Abmelden von Medien.
- Benutzerverwaltung: Neueintrag, Ändern und Löschen von Benutzern als zugelassene *smart*-Administratoren bzw. -Operatoren.

Die Benutzerschnittstelle wurde bedienerfreundlich gestaltet, d. h. die Hypertextdokumente enthalten selbsterklärende Texte und Beschreibungen, Formulare zur Erfassung von Konfigurationsdaten und Auswahlmenüs. Das Navigieren zwischen den verschiedenen Dokumenten wird durch sogenannte Hypertextlinks erleichtert. Aus diesem Grund wird auf eine detaillierte Beschreibung der Benutzerschnittstelle im Rahmen dieser Dokumentation verzichtet.

Auf die Datenbank greifen hauptsächlich der Volume Manager und die genannten CGI-Skripte zu. Abbildung 4.14 zeigt auch eine Verbindung des WWW-Servers zur Datenbank. Der WWW-Server verwendet die in der Datenbank abgelegten Benutzerinformationen für eine Zugriffskontrolle: CGI-Skripte und Hypertextdokumente von *smart* dürfen aus Sicherheitsgründen nur eingetragene Administratoren benutzen.

Kapitel 5

Restaurierung

Datensicherungen können Datenausfall oder -Verlust nicht verhindern. Sicherungskopien bieten jedoch die Möglichkeit, einen zerstörten Datenbestand wieder in einen früheren Zustand zu versetzen. Dieser Vorgang der Restaurierung von Daten soll im folgenden näher beschrieben werden.

smart enthält Software zur automatischen und regelmäßigen Datensicherung einer großen Anzahl von Rechnern auf zentralen Bandlaufwerken. Eine Datensicherung wird dabei von einem Programm (Volume Manager) weitgehend selbstständig durchgeführt. Ein Operator wird zum Wechseln von Bandmedien und zur Konfiguration des Systems benötigt. Daneben muß ein Operator nur noch bei Störungen eingreifen.

Eine Restaurierung läßt sich unter bestimmten Bedingungen ebenfalls automatisieren. Dazu gehören. . .

1. Benutzer, die eine Restaurierung anfordern, können die zu restaurierenden Daten genau beschreiben: Client, Filesystem, Verzeichnis und Dateinamen, an denen sich die Daten vor ihrem Verlust befunden haben.
2. Die Benutzer kennen den genauen Zeitpunkt, zu dem die Daten verloren gingen.
3. Für eine Restaurierung sind keine speziellen Arbeiten durch einen Administrator nötig.
4. Die zu restaurierenden Daten sollen an demselben Platz abgelegt werden, an dem sie sich vor deren Zerstörung befanden.

Leider treffen auf Restaurierungsanforderungen nur selten alle genannten Bedingungen zu. Die Gründe dafür sollen im folgenden den Bedingungen gegenübergestellt werden.

Durch eine umfangreiche Vernetzung vieler Rechner und eine Nutzung von Daten eines Rechners auf anderen Rechnern verschwimmen aus der Sicht der Benutzer die Grenzen zwischen den einzelnen Datenbeständen (Filesystemen). Nutzung von Speicherplatz erfolgt in vielen Applikationen transparent, d. h. aus der Sicht der Benutzer ist es unwichtig, wo sich die Daten tatsächlich befinden. Auf der Ebene des Betriebssystems (Shell) arbeiten Benutzer häufig mit sogenannten virtuellen oder symbolischen Pfaden beim Zugriff auf ihre Daten. Ein Pfad wie z. B. `/home/andreas/work/` oder `~andreas/work/` wird dabei vom Betriebssystem auf einen tatsächlichen Pfad `/export/disk1/home/andreas/work/`. Dies sind verschiedene Gründe, weshalb viele Benutzer nicht den Ort beschreiben können, an dem ihre Daten physikalisch gespeichert sind (widerspricht Bedingung 1).

Angaben zum tatsächlichen Speicherplatz werden aber für eine Restaurierung benötigt, da Sicherungskopien nur die tatsächlichen Pfade enthalten, nicht jedoch die symbolischen. Der Datensicherungsoperator hat keine Möglichkeit aus der Vielzahl von möglichen symbolischen Pfadangaben zu ermitteln, an welcher Stelle sich die Daten vor ihrem Verlust tatsächlich befunden haben. Als Hilfe für Benutzer und Administratoren ist im Anhang B ein Fragebogen zur Sammlung relevanter Informationen für eine Restaurierung aufgeführt.

Bevor eine Restaurierung erfolgen kann, muß ein zu restaurierender Datenbestand identifiziert und beschrieben werden. Ohne detaillierte Informationen über die verlorenen Daten, lassen sich entsprechende Sicherungskopien in einer großen Menge von Datensicherungen des Archivs nicht auffinden. Dennoch müssen vielfach Daten wiederhergestellt werden, die bereits vor längerer Zeit gelöscht oder zerstört wurden. Daneben können Daten verloren gehen, ohne daß dies von den Benutzern bemerkt wird. Benutzer können in diesen Fällen nur vage Angaben zum Zeitpunkt des Datenverlustes machen (widerspricht Bedingung 2). Auf dieses Problem geht Abschnitt 5.3.3 genauer ein.

Nach Ausfall und Reparatur einer Festplatte oder eines ganzen Rechners sind z. T. umfangreiche administrative Arbeiten zur Restaurierung der verlorenen Daten nötig (widerspricht Bedingung 3). Daneben gibt es Datenbestände, für die eine einfache Restaurierung einzelner Dateien nicht genügt. Bei Datenbankanwendungen können z. B. über die reine Restaurierung hinausgehende Arbeitsschritte notwendig werden (Einspielen der Daten, Index-Erstellung, etc.). Diese Arbeitsschritte muß meist ein Administrator der für einen Client zuständigen Abteilung durchführen. Eine Restaurierung von Daten kann also von der Applikation abhängig sein, mit der die Daten bearbeitet werden. Dagegen erfordert eine Wiederherstellung einzelner Benutzerdateien meist keine besonderen administrativen Arbeiten.

Schließlich kann es nötig werden, die zu restaurierenden Daten an einem anderen Platz oder auf einem anderen Client abzulegen (Ziel einer Restaurierung). Dies ist der Fall, wenn z. B. der ursprüngliche Rechner (Quelle der Originaldaten), auf dem sich die Daten befanden, nicht mehr existiert (widerspricht Bedingung 4). Falls sich Betriebssystem oder Rechnerarchitektur von Quelle und Ziel unterscheiden, können Unterschiede in der Anwendung von Restaurierungsprogrammen eine automatische Restaurierung verhindern. Eine Teilrestaurierung stellt einen anderen Grund dar: Nachdem ein Teil der Daten verfälscht wurde, arbeiteten die Benutzer für kurze Zeit mit dem fehlerhaften Datenbestand weiter. Nach der Entdeckung der Verfälschung sind die Originaldaten an einem anderen Ort zu restaurieren. Danach müssen alter und neuer Datenbestand durch die Benutzer verglichen und die verfälschten Daten ersetzt werden.

Die notwendigen Arbeitsschritte zur Restaurierung von Daten können im Einzelfall stark variieren. Aus diesem Grund läßt sich dieser Vorgang in der Praxis nicht vollständig automatisieren. Jedoch kann ein System zur zentralen Datensicherung die Arbeitsschritte, die ein Operator zur Restaurierung durchführen muß, auf ein Minimum reduzieren. Der folgende Abschnitt 5.1 beschreibt, welche Möglichkeiten *smart* dafür bietet.

5.1 Restaurierung mit *smart*

Jedem gesicherten Datencontainer wird vom Volume Manager eine eindeutige Volumenummer (Volume-ID) zugeordnet. Für eine Restaurierung ist daher die Kenntnis der Volumenummer eines Datencontainers notwendig.

Über ein Hypertextdokument (CGI-Skript) der *smart*-Benutzerschnittstelle kann ein Administrator nach den Volume-IDs der gewünschten Datencontainer suchen. Für diese Suche bietet die Benutzerschnittstelle mehrere Möglichkeiten zur Darstellung von Inhaltsverzeichnissen der Datenbank an. Diese Darstellung läßt sich nach verschiedenen Kriterien selektieren und sortieren.

Nachdem eine Volumenummer bekannt ist, folgt das Einlesen des Datencontainers (Volume) von den Speichermedien. Anhand der Volumenummer ermittelt der Volume Manager die Bandmedien, auf denen sich die Fragmente des gewünschten Datencontainers befinden. Medien, die nicht direkt über einen Bandlaufwerk oder Bandwechselsystem online verfügbar sind, fordert der Volume Manager über eine Console vom Operator unter Angabe der Mediennummern an. Sobald die Medien online zugreifbar sind, kann der Volume Manager mit dem Einlesen der Daten beginnen¹.

¹Genauer: sobald das erste Fragment eines Volumes online verfügbar ist. . .

Der Volume Manager überträgt diesen Datencontainer über die Device Manager der Clients an das gewünschte Ziel. Der Datencontainer steht damit am Ziel zum "Auspacken" bereit. Sein Inhalt kann in eine Datei oder "named pipe" geschrieben oder direkt an ein Systemkommando übergeben werden.

Die eigentliche Restaurierung beginnt mit der Wahl der Arbeitsweise für das "Auspacken". Der nächste Abschnitt 5.1.1 erläutert ein Vorgehen zur Restaurierung, das sich besonders für kleinere Datenbestände eignet.

5.1.1 Restaurierung einzelner Dateien/Verzeichnisse

Eine Restaurierung kleiner Datenbestände läßt sich zum Teil automatisieren. Es müssen dabei die auf Seite 47 aufgezählten Bedingungen 3 und 4 zutreffen. Dazu fordert ein Administrator vom Volume Manager über ein Hypertextformular eine Liste aller in einem Datencontainer enthaltenen Dateien und Verzeichnisse an. Diese Liste muß der Volume Manager aus dem Inhaltsverzeichnis des jeweiligen Datencontainers erstellen. Sie ist aufgrund ihres Umfangs nicht in der Datenbank des Volume Managers enthalten. Der Volume Manager holt den Datencontainer von den Bandmedien und übermittelt ihn an einen Client. Über diesen Client startet der Volume Manager ein entsprechendes Restaurierungsprogramm (-Skript) und läßt sich eine Liste der in dem Datencontainer enthaltenen Dateien ausgeben. Diese Liste erhält nun der Administrator über die Benutzerschnittstelle als Hypertextdokument angezeigt.

Der Administrator wählt die für eine Restaurierung in Frage kommende Verzeichnisse bzw. Dateien aus. Über ein weiteres Hypertextformular müssen zusätzlich noch Angaben zum Ziel der Restaurierung gemacht werden:

- Auf welchem Rechner sollen die Daten ausgepackt werden?
- In welches Verzeichnis sind die Daten zu restaurieren?

Die Restaurierbarkeit eines Datencontainers auf einem bestimmten Zielrechner (-Betriebssystem) ist abhängig vom Datensicherungsprogramm, das den Datencontainer erzeugt hat. Diese Datencontainer sind aufgrund unterschiedlicher Formate (Typen) nicht beliebig zwischen verschiedenen Rechnern bzw. Betriebssystemen austauschbar. Sollten sich aus der Wahl des Ziels Inkompatibilitäten für das Format ergeben, meldet der Volume Manager diesen Umstand dem Administrator, so daß dieser ein alternatives Ziel wählen kann.

Aufruf und Bedienung des Restaurierungsprogramms (-Skripts) wird bei diesem Verfahren zur Restaurierung vom Volume Manager selbstständig abgewickelt.

5.1.2 Restaurierung größerer Datenbestände

Zur Durchführung einer manuellen Restaurierung benötigt ein Administrator in der Regel root-Privilegien auf dem Client, auf dem die Daten "ausgepackt" werden sollen. Über ein Hypertextformular gibt der Administrator die gewünschte Volumenummer und eine "Pipe-Resource" als DM-Objekt an. Der Volume Manager wird daraufhin den Inhalt des gewünschten Datencontainers über die angegebene "named pipe" eines Device Managers bereitstellen.

Der Administrator kann daraufhin den Inhalt des Datencontainers mit einem entsprechenden Restaurierungsprogramm verarbeiten (restaurieren). Aufruf und Bedienung des Restaurierungsprogramms ist dabei Aufgabe des Administrators.

5.2 Restaurierung ohne *smart*

Alle bisher beschriebenen Restaurierungsverfahren benutzen den Service des Volume Managers. Für den Fall, daß der Server der zentralen Datensicherung ausfällt, die Datenbank oder Software des Volume Manager verloren gehen, können die Dienste des Volume Managers nicht mehr benutzt werden. Es müssen deshalb Möglichkeiten existieren, die eine Restaurierung auch ohne die *smart*-Software erlauben.

5.2.1 Zerstörte *smart*-Datenbank

Mit Hilfe eines im späteren Abschnitt 6.2.3.2 erwähnten Hilfsprogramms lassen sich regelmäßig Kopien des Datenbankinhalts erstellen. Diese Kopien können nach einem Verlust der *smart*-Datenbank für eine Neugenerierung der Datenbank verwendet werden. Zur Restaurierung bzw. Generierung muß an dieser Stelle auf den erwähnten Abschnitt, sowie auf die Dokumentation des Datenbanksystems verwiesen werden.

5.2.2 Rekonstruktion von Inhaltsverzeichnissen

Für den Fall, daß für eine Restaurierung die in der *smart*-Datenbank abgelegten Inhaltsverzeichnisse nicht zur Verfügung stehen, lassen sich diese Verzeichnisse teilweise aus dem Inhalt der Datensicherungsmedien rekonstruieren. Die wichtigsten Informationen sind in den Band- und Filelabel der Sicherungsmedien enthalten. Diese Label bestehen jeweils aus einem Datenblock von 1024 Byte Länge und stehen jeweils am Anfang eines jeden

Mediums (Bandlabel, File #0) bzw. vor der Aufzeichnung eines Fragments (Filelabel, Block #0 jedes Files). Diese Datenblöcke enthalten einen Informationstext der folgenden Form:

```
Format:          EXA5F
Label:           762
File:            9
Created:         Fri Nov 22 13:37:10 1996
Host:            ako
Device:          /dev/scsi/rexabyte
Program Version: dm-A4.31+03[t] 1996/10/22 14:10:03
Info:
  Volume-ID=220841 Fragment-Offset=0
  Client=zdi Filesystem=/export/home
```

Informationen zur Verwendung im Textfeld "Info" müssen dem jeweiligen Device Manager vor einem Schreiben von Daten auf Band mitgeteilt werden. Dies erfolgt über die Option "info=..." der DM-Kommandos `assign` und `setoutput` (siehe Anhang A.1.3). Bei einer Datensicherung über *smart* werden vom Volume Manager entsprechende `info`-Texte verwendet.

Über die in jedem Label enthaltenen Informationen können mit Hilfe von einfachen Shell-Skripten neue Inhaltsverzeichnisse erstellt werden. Da diese Aktion sehr zeitaufwendig sein wird, ist es sinnvoller aus der Menge der Datensicherungen zunächst die Datensicherungen der *smart*-Datenbank zu suchen und diese zu restaurieren. Anhand der wiederhergestellten Datenbank und ihrer Inhaltsverzeichnisse können nun alle anderen Datensicherungen gefunden werden.

5.2.3 Restaurierung über Device Manager

Ein Zugriff auf gesicherte Daten ohne die Hilfe von *smart* setzt voraus, daß bekannt ist, wie die einzelnen Fragmente eines Datencontainers auf die Sicherungsmedien verteilt sind.

Mit dem DM-Benutzerinterface `dmcp` ist ein relativ einfacher Zugriff auf die Daten eines Volumes möglich. Beispiel: ein Administrator hat den Speicherort der Fragmente eines Datencontainers ermittelt und die notwendigen Speichermedien bereits in das Magazin eines Bandwechselsystems eingelegt. Die Angaben zu diesem Datencontainer sind als Tabelle zusammengefaßt:

Fragment	Offset in Byte	Länge in Byte	Filenr.	Medienlabel	Slot
1	0	5242880 (5m)	12	EXA5F-762	1
2	5242880 (5m)	52428800 (50m)	49	EXA5F-762	1
3	57671680 (55m)	21725184 (Rest)	2	EXA5F-763	2
Gesamtlänge: 79396864					

Mit diesen Angaben läßt sich ein `dmcp`-Kommando bilden². Im Beispiel ist das Bandlaufwerk über die *Device-Resource* "inf:Dexa5f" ansprechbar. Als Ziel der Datenübertragung sollen die Daten auf einem anderen Rechner an eine neue *Pipe-Resource* "zdi:P/tmp/restore" übergeben werden:

```
dmcp inf:Dexa5f label=762 slot=1 file=12 limit=5m \
    inf:Dexa5f label=762 slot=1 file=49 limit=50m \
    inf:Dexa5f label=763 slot=2 file=2 \
    -- zdi:P/tmp/restore create
```

Die einzelnen Zeilen des `dmcp`-Kommandos enthalten jeweils DM-Objekte und -Optionen der DM-Kommandoschnittstelle. Anhang A.3 erläutert die Kommandosyntax eingehender.

Nach dem Start des gezeigten `dmcp`-Kommandos, kann der Administrator auf dem Rechner `zdi` den Inhalt des Datencontainers aus der angegebenen "named pipe" auslesen und verarbeiten.

5.3 Schwierigkeiten einer Restaurierung

Die folgenden Abschnitte beschreiben verschiedene Schwierigkeiten einer zentralen Datensicherung, an denen eine Restaurierung scheitern kann.

Neben den hier gezeigten Maßnahmen und Vorkehrungen zur Vermeidung dieser Probleme, enthält [IT 96] einen umfangreichen Katalog mit Maßnahmen zum Schutz von Daten und hilft bei einer Erstellung von Notfallplänen.

5.3.1 Hardware

Voraussetzung für ein zentrales Datensicherungssystem ist eine funktionierende Hardware der beteiligten Instanzen. Dazu sind zu zählen:

- Server, auf denen die Software des Volume Managers, des Datenbanksystems und des WWW-Servers läuft;

²Voraussetzung: dem Administrator ist Syntax und Semantik von DM-Kommandos bekannt.

- Netzwerk, das Clients und Server miteinander verbindet;
- Clients, deren Datenbestände zu sichern sind;
- Clients, über deren Device Manager Zugriffe auf die Bandlaufwerke erfolgen;
- Bandlaufwerke;
- Bandmedien.

Es ist offensichtlich, daß Datensicherungen gestört werden können, sobald eine oder mehrere der genannten Komponenten ausfallen. Die *smart*-Software wurde so konzipiert, daß Störungen eines Client sich nicht auf Datensicherungen anderer Clients auswirken, sofern diese nicht ebenfalls direkt von einer Störung betroffen sind. Bandmedien wurden mit aufgezählt, da eine Restaurierbarkeit von Daten in erster Linie von der Güte der auf ihnen gespeicherten Daten abhängt.

Weniger offensichtlich sind indirekte Gefahren für eine Restaurierbarkeit gesicherter Daten. Beispiele dazu:

- Für eine zentrale Datensicherung steht ein Bandlaufwerk hoher Leistungsfähigkeit und Kapazität zur Verfügung. Auch wenn dieses Bandlaufwerk ausreichend dimensioniert wurde, stellt es für die Restaurierbarkeit eine Gefahr dar. Nach einem Defekt dieses Bandlaufwerks können Restaurierungen erst wieder nach der Beschaffung eines Ersatzlaufwerks durchgeführt werden.

Eine wirksame Maßnahme zur Erhöhung der Verfügbarkeit ist in diesem Fall die Bereitstellung eines zweiten Bandlaufwerks als Reserve.

- Obwohl in einer zentralen Datensicherung ein zweites Bandlaufwerk als Reserve zur Restaurierung bereitsteht, kann eine Restaurierbarkeit dennoch gefährdet sein. Aufgrund von Toleranzen oder äußerlich nicht erkennbarer Unterschiede der Hardware von Bandlaufwerken, kann es sein, daß Bandmedien nur auf dem einen Bandlaufwerk gelesen werden können, auf dem diese beschrieben wurden. Als Beispiel wurden in der Vergangenheit für Laufwerke vom Typ Exabyte 8500 Erweiterungen angeboten, die eine Kapazitätserhöhung der Speichermedien mit Hilfe von Datenkompression erreichen. Jedoch waren die verwendeten Aufzeichnungsformate bzw. Kompressionsalgorithmen der einzelnen Hersteller nicht miteinander kompatibel. Ein Austausch von Medien zwischen unterschiedlich ausgerüsteten Bandlaufwerken war damit unmöglich.

Als Maßnahme zur Sicherstellung einer Restaurierbarkeit von Daten müssen alle eingesetzten Bandlaufwerke durch die Administratoren

geprüft werden. Es ist sicherzustellen, daß Medien, die auf einem Laufwerk beschrieben wurden, auf allen anderen dafür vorgesehenen Laufwerken gelesen werden können. Nach einer Ersatzbeschaffung oder Ergänzung von Hardware sind diese Tests zu wiederholen.

- Toleranzen eines Laufwerks bei der Aufzeichnung und dem Lesen von Bandmedien können sich im Laufe der Zeit verändern. Dadurch kann zum einen eine Austauschbarkeit von Medien zwischen einzelnen Bandlaufwerken gefährdet sein. Zum anderen kann unter Umständen sogar das Lesen älterer Daten unmöglich werden, die auf demselben Laufwerk beschrieben wurden.

Eine Maßnahme gegen eine Veränderung von Toleranzen ist eine regelmäßige Wartung der Laufwerke nach Angaben der Hersteller. Dazu gehören die Einhaltung von Reinigungszyklen und Inspektionen durch eine Fachwerkstatt. Zusätzlich sollten alte Medien aus einem Bandarchiv regelmäßig auf Lesbarkeit geprüft werden.

- Neben der schleichenden Veränderung von Toleranzen eines Laufwerks, verändern sich ebenso die in einem Archiv gelagerten Bandmedien. In der Regel schreiben Bandlaufwerke Daten mit einem magnetischen Verfahren auf die Medien. Der Inhalt eines Magnetbands besteht also aus einem magnetischen Muster. Dieser Inhalt kann zum einen durch direkte physikalische Einwirkungen zerstört werden:
 - Staub, zerkratzt die Oberfläche des Bandmediums und des Schreib-/Lesekopfes eines Bandlaufwerks während des Lesens der Daten.
 - Feuchtigkeit, bildet zusammen mit Staub einen Schmierfilm auf der Oberfläche des Mediums.
 - Hitze (Feuer), Gehäuse und Bandträgermaterial können sich verziehen oder schmelzen.

Neben den direkten Einwirkungen auf das Bandmaterial wird ein magnetisches Muster im Laufe der Zeit von selbst schwächer oder wird durch fremde Muster überlagert. Dieser Vorgang verstärkt sich mit zunehmender Lagertemperatur.

Als Maßnahme zur Sicherstellung der Lesbarkeit alter Archivmedien ist ein regelmäßiges Kopieren der wichtigsten Daten auf neue Bänder nötig. Eine Wiederverwendung alter Archivbänder zur Datensicherung ist nicht empfehlenswert, da deren Zuverlässigkeit aufgrund der beschriebenen Gefahren mit zunehmendem Alter nachläßt. Zusätzlich sollten Aufzeichnungen in einem dafür geeignetem und entsprechend geschütztem Raum (Bandarchiv) gelagert werden.

5.3.2 Software

Eine erfolgreiche Restaurierung kann aufgrund von Softwarefehlern erschwert oder sogar unmöglich gemacht werden. Es folgen ausgewählte Beispiele, der in der Praxis vorkommenden Probleme:

- Die zur Datensicherung verwendete Software muß in der Lage sei, zu sichernde Daten als Datencontainer zu “verpacken” und auf Sicherungsmedien zu schreiben. Ebenso müssen diese Datencontainer bei einer Restaurierung von einem Bandlaufwerk wieder auf einem Client zum “Auspacken” bereitgestellt werden. Die korrekte Funktion des Systems zur zentralen Datensicherung muß gewährleistet sein.
- Betriebssysteme benutzen für den Zugriff auf Peripheriegeräte sogenannte Device-Treiber. Diese Treibersoftware enthält in vielen Betriebssystemen versteckte Fehler oder Schwächen.

Die Device Manager der *smart*-Software verwenden Filenummern zur Auswahl einzelner Aufzeichnung von Bandmedien. Es kommt vor, daß die Angabe von Filenummern und Durchführung von Bandpositionierungsoperationen in den Device-Treibern der Betriebssysteme fehlerhaft sind. So “verzählen” sich z. B. die Betriebssysteme SunOS 4.1.1 und Linux bis Version 2.0.3 in der Filenummer bei bestimmten Bandoperationen.

- Abschnitt 5.1.1 erwähnt mögliche Inkompatibilitäten falls Clients von Quelle und Ziel eines Datencontainers unterschiedliche Betriebssystemversionen oder unterschiedliche Implementationen von Sicherungsprogrammen verwenden. Ebenso können Unterschiede im Format der Datencontainer eine Restaurierung unmöglich machen.

Zum Beispiel verwendet das SunOS-Sicherungsprogramm `dump` für im Datencontainer enthaltene Inhaltsverzeichnisse eine Struktur, die auf 512 Byte großen Verzeichniseinträgen basiert. Das `dump`-Programm des Betriebssystems HP-UX verwendet dagegen 1024 Byte große Einträge. Unglücklicherweise ist das dazugehörige Restaurierungsprogramm `restore` nicht in der Lage eine nicht vom eigenen System verwendete Größe zur verarbeiten. Folglich können Sicherungen eines HP-Rechners nicht auf einem SunOS-Rechner restauriert werden und umgekehrt.

- Selbst nachdem eine Restaurierung erfolgreich durchgeführt werden konnte, bleibt der Datenbestand gefährdet. Ein fehlerhaft arbeitendes Anwenderprogramm, das die ursprünglichen Daten zerstörte, “bedroht” selbstverständlich auch weiterhin die restaurierten Daten.

Deshalb sollte ein Datensicherungsoperator vor einer Restaurierung den Grund für einen Datenverlust ermitteln. Auf diese Weise läßt sich das fehlerhafte Programm rechtzeitig korrigieren oder ersetzen. Ein wiederholter Verlust des Datenbestands wird damit verhindert.

Als Maßnahme auf Seiten der Software, zur Sicherstellung einer hohen Qualität von Datensicherungen, kann hier nur eine Stichprobenkontrolle aufgeführt werden. In regelmäßigen und unregelmäßigen Abständen werden willkürlich ausgewählte Datenbestände restauriert. Diese Restaurierung muß dabei an einem freien und ungenutzten Speicherplatz erfolgen. Danach sind die restaurierten Daten mit den ursprünglichen Daten zu vergleichen. Sofern die Originaldaten seit der Erstellung der Sicherungskopien unverändert geblieben, müssen die beiden Datenbestände identisch sein.

5.3.3 Suche im Archiv

Bei Einsatz einer inkrementellen Datensicherungsmethode benötigt ein Administrator zur Restaurierung verlorener Daten genaue Informationen, wann diese Daten zuletzt verändert wurden. Lassen sich diese Zeitangaben nicht ermitteln, können die gewünschten Datencontainer nur durch eine aufwendige Suche im Archiv gefunden werden. Eine solche Suche soll an einem typischen Beispiel erläutert werden: Ein Benutzer meldet sich beim Administrator der zentralen Datensicherung und bittet um die Restaurierung von Daten. Da der Benutzer die Daten seit einiger Zeit nicht mehr verwendet hat, weiß er nicht, wann diese Daten verloren gingen. Zudem kann er nur ungenaue Angaben über den Zeitpunkt der letzten Veränderung machen. Die Datensicherung wurde mit der in Abschnitt 3.3 besprochenen Sequenz 1 durchgeführt: 05555 35555. Eine Sequenz erstreckt sich also über 10 einzelne Sicherungen.

Der Administrator hat nun das Problem, diejenigen inkrementellen Sicherungen (oder Gesamtsicherung) herauszusuchen, welche die aktuellste Version der verlorenen Daten enthalten. Dies stellt ein Problem dar, da üblicherweise keine vollständigen Inhaltsverzeichnisse aller während einer Datensicherung kopierten Dateien existieren. Ein solches Dateiverzeichnis ist in der Regel viel zu groß und wird deshalb häufig von Systemen zur zentralen Datensicherung nicht erstellt. Werden Informationen über einzelne gesicherte Dateien, wie in dem genannten Beispiel benötigt, so muß die entsprechende Sicherung vom Bandmedium gelesen werden. In der Sicherung selbst ist ein Inhaltsverzeichnis der darin enthaltenen (gesicherten) Dateien vorhanden.

Zur Restaurierung der Daten aus dem oben angeführten Beispiel, geht der Administrator schrittweise vor:

1. Suche nach einer Gesamtsicherung (dump level 0), die zeitlich in der Nähe des vom Benutzer vermuteten Zeitpunkts des Datenverlustes erzeugt wurde. Diese Sicherung prüfen, ob die gewünschten Daten enthalten sind. Wenn nicht, dann Schritt 1 mit anderer Gesamtsicherung wiederholen.
2. Wurde eine Gesamtsicherung mit den gewünschten Daten gefunden, muß geprüft werden, ob die nachfolgende Gesamtsicherung eine aktuellere Version der Daten enthält. Wenn Ja, dann Schritt 2 mit dieser Gesamtsicherung wiederholen.
3. Es folgt die Suche nach einer inkrementellen Sicherung, die aktuellere Versionen der Daten enthalten könnte, als die der Gesamtsicherung. Eine optimale Suchreihenfolge ist vom Einzelfall und der verwendeten Sicherungssequenz abhängig. In diesem Fall sollte zunächst die Sicherung des dump level 3 geprüft werden. Enthält diese Sicherung die gesuchten Daten, müssen die nachfolgenden Level ebenfalls geprüft werden. Andernfalls sind die unmittelbar vorhergehenden level-5-Sicherungen zu prüfen.

Es ist absehbar, daß die Suche nach den gewünschten Daten zur Restaurierung sehr zeitaufwendig sein kann. Ohne detaillierte Informationen über die verloren Daten, lassen sich entsprechende Sicherungskopien in einer großen Menge von Datensicherungen des Archivs nur schwer auffinden. Deshalb liegt es im Interesse von Benutzer und Administrator möglichst genaue Angaben über die zu restaurierenden Daten zu sammeln.

Bei einer Suche im Archiv können zusätzliche Informationen hilfreich sein und den Suchraum einschränken:

- Entstehungszeit der Daten: Diese Angabe erspart dem Operator eine Suche nach den gewünschten Daten in Sicherungen vor diesem Zeitpunkt.
- Änderungshäufigkeit: Datenbestände, die häufig (z. B. täglich) verändert werden, sind vermutlich in jeder Sicherungskopie enthalten. Dagegen finden sich selten geänderte Daten nur in wenigen inkrementellen Sicherungskopien. Eine Suche im Archiv nach diesen Daten ist daher aufwendiger.

Zur Ermittlung von notwendigen Informationen zur Restaurierung bietet sich der bereits erwähnte Fragebogen aus Anhang B an.

Kapitel 6

Aufbau und Betrieb von *smart*

Dieses Kapitel beschreibt die Implementation des *smart*-Softwarepakets, seine Installation und Konfiguration, sowie Aspekte des Betriebs in der zentralen Datensicherung.

6.1 Device Manager

6.1.1 Implementierung

In [DM, Kapitel 5] werden Arbeitsweise und interne Strukturen des Device Managers beschrieben. Da der Device Manager im Rahmen einer Studienarbeit entstand, soll an dieser Stelle für eine Beschreibung seiner Implementierung auf die zitierte Studienarbeit verwiesen werden.

6.1.2 Installation

Bei der Entwicklung des Device Managers wurde Wert auf eine hohe Portabilität der Software gelegt. Sie sollte daher auf allen gebräuchlichen UNIX-Versionen und Rechnerplattformen unverändert einsetzbar sein. Die Software des Device Managers ist als Unterverzeichnis `dm` im *smart*-Softwarepaket enthalten. Die Datei `dm/README` enthält eine jeweils aktuelle Liste von Betriebssystemen und Hardware. Auf den in dieser Liste genannten Systemen wurde der Device Manager getestet. Sie gibt zusätzlich Hinweise zur Installation und Portierung auf weitere Systeme.

Allgemein wurden zwei Möglichkeiten zur Installation der Software vorgesehen:

- Auf einem Client wird nur der Device Manager benötigt, nicht jedoch der Volume Manager und die CGI-Skripte der *smart*-Benutzerschnittstelle. Daher genügt es, die unten beschriebenen Kommandos in dem erwähnten Unterverzeichnis `dm` auszuführen.
- Auf dem Server der zentralen Datensicherung werden in der Regel alle Teile des *smart*-Softwarepakets eingesetzt. Deshalb sind die Kommandos im Hauptverzeichnis des Softwarepakets auszuführen.

Die Übersetzung und Installation besteht aus folgenden Shell-Kommandos:

```
sh configure
make depend
make all
make check
make install
```

Das Kommando “`sh configure`” akzeptiert einige Optionen, mit denen der Ablauf von Übersetzung und Installation verändert werden kann. Eine Liste aller erlaubten Optionen erhält man mit...

```
sh configure --help
```

Es ist empfehlenswert Konfigurationsfiles, Programme und Skripte des Device Managers in ein eigenes Verzeichnis zu legen. Mit dem Kommando...

```
sh configure --prefix=/home/backup/smart
```

... wird die Software unterhalb des Verzeichnisses `/home/backup/smart` installiert. Als Prefix für den Pfadnamen ist `/usr/local` als Standardwert vorgesehen. Detaillierte Informationen über “`configure`” sind in den Dateien “`README`” und “`INSTALL`” enthalten.

6.1.3 Konfiguration

Dieser Abschnitt beschreibt notwendige Vorbereitungen zum Betrieb eines Device Managers als Client der zentralen Datensicherung unter *smart*.

6.1.3.1 Automatischer Programmstart des Device Managers

In [DM, Kapitel 4.2] wird beschrieben, wie das Programm `dm` des Device Managers während eines Bootvorgangs des Betriebssystems gestartet werden kann. Ein `dm`-Programm, auf diese Weise gestartet, läuft ständig und wartet auf Kommandos eines Benutzers oder des Volume Managers.

Zur Schonung der Ressourcen eines Clients wurde deshalb zusammen mit der Entwicklung von *smart* eine weitere Möglichkeit zum Starten eines Device Managers geschaffen. Das *dm*-Programm wird dabei vom "Internet Super-Server" *inetd* bei Bedarf gestartet¹. Der Device Manager bearbeitet danach die an ihn gegebenen Kommandos. Nachdem er alle Kommandos bearbeitet hat und alle Verbindungen zu den Benutzern geschlossen wurden, beendet er sein laufendes Programm nach einer kurzen Wartezeit selbst. Damit werden alle von ihm belegten Ressourcen (z. B. Arbeitsspeicher) freigegeben. Sobald ein neues Kommando für den Device Manager eintrifft, startet *inetd* das Programm *dm* erneut.

Zum Starten des DM über *inetd* muß erstens in die Datei */etc/inetd.conf* folgende Zeile eingetragen werden²:

```
dm stream tcp wait root /home/backup/sbin/smart/dm dm -l
```

Je nach verwendeten Pfadnamen als Prefix ist diese Zeile entsprechend anzupassen.

Zweitens muß in */etc/services* ebenfalls eine Zeile eingefügt werden³:

```
dm 511/tcp # Device Manager
```

Fehlermeldungen und Warnungen des Device Managers werden über den Betriebssystemdienst Syslog protokolliert⁴. Mit einem Eintrag in der Konfigurationsdatei */etc/syslog.conf* von...

```
local4.notice /var/adm/dm-log
```

werden alle Protokollausgaben⁵ des DM in die Datei */var/adm/dm-log* geschrieben. Alternativ kann auf einem Client mit der Zeile

```
local4.notice @serverhost
```

erreicht werden, daß DM-Protokolle an einen anderen Rechner mit Namen "serverhost" geschickt und diese dort verarbeitet werden. Sinnvollerweise verwendet man als Hostname anstelle von "serverhost" einen Server der zentralen Datensicherung. Die in den Beispieleinträgen für die Datei *syslog.conf* verwendete Syslog-Facility *local4* ist im Quellcode des Device Managers voreingestellt. Die Übermittlung von Protokollen an einen

¹Siehe UNIX-Manualseite zu *inetd(8)*.

²Siehe UNIX-Manualseite zu *inetd.conf(5)*.

³Siehe UNIX-Manualseite zu *services(5)*.

⁴Siehe UNIX-Manualseite zu *syslog(8)* bzw. *syslogd(8)* und *syslog.conf(5)*.

⁵Genauer: alle Protokollausgaben mit Syslog-Priorität *notice* und höher.

zentralen Server erleichtert in Fehlersituationen eine Problemlösung durch die Administratoren der zentralen Datensicherung.

Für den Fall, daß der Systemdienst `syslog` nicht verwendet werden kann/soll, ist aus der Zeile, die in `inetd.conf` für den Device Manager eingetragen wurde, die Option “-1” zu entfernen.

Nach Änderung der Dateien `inetd.conf`, `services` und `syslog.conf` sind die Daemon `inetd` und `syslogd` neu zu starten bzw. zu initialisieren. In der Regel erfolgt dies durch Hangup-Signale, die an diese Daemon gesendet werden:

```
kill -HUP <pid-of-inetd>
kill -HUP <pid-of-syslogd>
```

Für alle bisher gezeigten Konfigurationsbeispiele gilt: Syntax und Semantik der Konfigurationsdateien, sowie verwendete Pfadnamen, stammen aus gebräuchlichen UNIX-Betriebssystemen. Es ist möglich, daß einige UNIX-Systeme entsprechend angepaßte Angaben verlangen⁶.

6.1.3.2 Konfigurationsdatei `dm.conf`

Aufbau und Format der Konfigurationsdatei `dm.conf` ist in [DM, Kapitel 4.3] und im Anhang A.1.4 beschrieben. An dieser Stelle werden nur die notwendigen Bestandteile dieser Konfigurationsdatei für einen Betrieb als Client unter *smart* als Beispiel gezeigt.

```
# dm.conf
default rootonly=true chdir=/home/backup/smart
serverhost
wwwhost
```

Diese Angaben beschränken den Zugriff auf Funktionen des Device Managers auf Programme und Benutzerschnittstellen, die auf den Rechnern “serverhost” und “wwwhost” mit root-Privilegien laufen. “serverhost” sollte der Name des Rechners sein, auf dem der Volume Manager installiert wird. “wwwhost” sollte dagegen der Name des WWW-Servers sein⁷. Diese Zeilen sind unter dem Dateinamen

```
/home/backup/smart/etc/dm.conf
```

⁶Bei einigen `syslog`-Implementationen sind die einzelnen Spalten eines `syslog.conf`-Eintrags durch Tabulatorzeichen anstelle von Leerzeichen zu trennen.

⁷`serverhost` und `wwwhost` können dabei identisch sein, falls Volume Manager und WWW-Daemon auf demselben Rechner laufen.

abzulegen⁸. Je nach verwendetem Prefix ist der Pfad dieser Datei entsprechend anzupassen.

6.1.3.3 Konfigurationsdatei `dev.conf`

Die Konfigurationsdatei `dev.conf` wird unter *smart* nur auf einem Rechner benötigt, der über ein Bandlaufwerk verfügt. [DM, Kapitel 4.3] enthält ein Beispiel und Erläuterungen für eine solche Datei. Desweiteren beschreibt Anhang A.1.4 ihre Syntax und Semantik. In diesem Anhang ist ein umfangreiches Beispiel einer solchen Konfigurationsdatei enthalten. Die Datei "`dev.conf`" wird vom Device Manager in demselben Verzeichnis erwartet, in dem er "`dm.conf`" gefunden hat.

6.2 Volume Manager

6.2.1 Implementierung

Die Benutzerschnittstellen des Volume Managers wurden als CGI-Skripte (dynamische Hypertextdokumente eines WWW-Server) gestaltet. Zu diesem Zweck kommt häufig die Programmiersprache Perl zum Einsatz. Es lag daher nahe, Perl als Implementierungssprache auch für den Volume Manager zu verwenden. Vorzüge von Perl Version 5:

- Perl ist eine Sprache, in der sowohl objektorientierte als auch betriebssystemnahe Programme leicht zu schreiben sind.
- Perl unterstützt einen modularen Aufbau der Software.
- Es existiert eine umfangreiche Softwarebibliothek. Dazu gehören Programmmodule zur Einbindung verschiedener Datenbanken und Software zur einfachen Generierung von Hypertextdokumenten.
- Der Perlinterpret bzw. -Compiler ist auf (fast) allen UNIX-Betriebssystemen und auf einigen Nicht-UNIX-Systemen verfügbar.

Die Wahl von Perl als Implementationssprache erleichtert eine Portierung von *smart* auf neue Betriebssysteme, sofern Portierungsarbeiten überhaupt notwendig sein sollten.

Die Software des Volume Managers besteht aus einer Anzahl von Perlmodulen. Diese Modularisierung macht eine rasche Anpassung der Software möglich z. B. nach Wechsel des eingesetzten Datenbanksystems.

⁸Annahme: Der Device Manager wurde unter Angabe eines Prefix von `/home/backup/smart` übersetzt und installiert.

Eine detaillierte Beschreibung der internen Arbeitsweise und Struktur des Volume Managers würde den Rahmen dieser Dokumentation übersteigen. Die grobe Struktur besteht aus einem mehrschichtigen Aufbau aus einzelnen Modulen. Ihre Aufgaben sind:

DM-Verbindung: Die unterste Schicht ist zuständig für Aufbau eines Kommandokanals vom Volume Manager zu den einzelnen Device Managern der Clients und übermittelt DM-Kommandos. Dieses Modul sammelt Meldungen und Antworten (replies) der Device Manager und übergibt sie der nächsten Schicht.

DM-Kommando: Diese Schicht bearbeitet einzelne DM-Kommandos: Zusammenstellen neuer Kommandozeilen, Auswerten von Rückmeldungen des Device Managers. Die gesammelten und aufbereiteten Ergebnisse einzelner DM-Kommandos werden an die nächste Schicht gemeldet.

VM-Task: Eine Task des Volume Managers besteht aus einer abgeschlossenen Aufgabe. Eine solche Aufgabe ist z. B. Starten eines Datensicherungsprogramms auf einem Client und Übertragung des Datencontainers auf ein Ziel. Eine Task verwendet zur Durchführung ihrer Aufgaben DM-Kommandos und kann bei komplexeren Aufgaben weitere Tasks (subtasks) starten, die dann eine Teilaufgabe bearbeiten.

VM-Scheduler: Kernstück des Volume Manager – Koordination mehrerer Tasks über Ereignisse. Jede Task löst verschiedene Operationen aus und wartet danach auf deren Ergebnisse. Während dieser Wartezeit kann sich der Scheduler anderen Tasks "widmen". Sobald ein Ereignis (Ergebnis) für eine Task eintrifft, setzt der Scheduler die Bearbeitung der betreffenden Task fort. Damit ähnelt die Arbeitsweise des Schedulers dem eines Multitasking-Betriebssystems.

VM-Planer: Ist eine eigenständige Task, die beim Programmstart des Volume Managers initiiert wird. Aufgrund der in der Datenbank abgelegten Informationen und Regeln startet der VM-Planer neue Tasks zur Durchführung von Datensicherungen und koordiniert dabei die Nutzung eines Cache, Fragmentierung von Datencontainern, Multiplexen von Fragmenten, Zugriff auf Bandlaufwerke usw.

VM-Interface: Über ein Interface nehmen CGI-Skripte Verbindung mit dem Volume Manager auf und übermitteln Anfragen und Aufträge von Benutzern. Dabei werden z. T. neue VM-Tasks gestartet oder Informationen vom Volume Manager abgefragt.

Eine Task des Volume Managers besteht aus einer Sammlung von Unterprogrammen, die in einer Tabelle eingetragen wurden. Jede Task-Tabelle

stellt einen deterministischen endlichen Automaten dar⁹. Ereignisse, auf die eine Task wartet, bilden dabei die Eingabesprache des Automaten. Der Scheduler übergibt ein eingetroffenes Ereignis (z. B. Rückmeldung eines DM) dem für dieses Ereignis zuständigen Automaten. Dabei wird dann ein in der Automatentabelle einer Task eingetragenes Unterprogramm aufgerufen.

Durch die Realisierung von Tasks als endliche Automaten und einer Anzahl von Unterprogrammen läßt sich die Software des Volume Managers leicht erweitern. Zusätzlich konnte mit Mitteln der Programmiersprache Perl eine Sicherung gegen Fehler eingebaut werden. Laufzeitfehler, die während der Bearbeitung einer Task auftreten, führen lediglich zum Abbruch der fehlerhaften Task. Ein Absturz des Volume Managers ist dabei nicht möglich. Ein solcher Fehler hat also keine direkten Auswirkungen auf andere von diesem Fehler unabhängige Tasks.

6.2.2 Installation

Der Volume Manager wird wie der Device Manager durch ein `configure`-Kommando mit entsprechenden Parametern und einer Reihe von `make`-Anweisungen vorbereitet und installiert. Die Angaben aus Abschnitt 6.1.2 gelten sinngemäß auch für den Volume Manager. Die Software des Volume Managers befindet sich dabei im Unterverzeichnis `vm` des *smart*-Softwarepakets.

6.2.3 Konfiguration

Dieser Abschnitt beschreibt notwendige Vorbereitungen zum Betrieb von Volume Manager, Datenbank und *smart*-Benutzerschnittstellen.

6.2.3.1 Automatischer Programmstart des Volume Managers

Der Volume Manager besteht aus einem ständig laufenden Perl-Skript. Das Skript bzw. Programm `vm` muß dazu auf dem Server der zentralen Datensicherung bei jedem Reboot des Betriebssystems mit `root`-Privilegien automatisch gestartet werden. Vor einem "shutdown" des Servers muß der Volume Manager "sauber" beendet werden. Diese Aufgabe übernimmt ein kleines Shell-Skript, das an entsprechenden Stellen des Betriebssystems "eingebaut" werden muß.

⁹Deterministische endliche Automaten werden in [AHO 88] ausführlich behandelt.

Das folgende Beispiel zeigt eine Möglichkeit, wie das Betriebssystem SunOS ab Version 5.0 für einen automatischen Start des Volume Managers vorbereitet werden kann¹⁰.

```
cp -p vm_daemon.sh /etc/init.d
cd /etc/rc3.d; ln -s ../init.d/vm_daemon.sh S99vm
cd /etc/rc2.d; ln -s ../init.d/vm_daemon.sh K01vm
cd /etc/rc1.d; ln -s ../init.d/vm_daemon.sh K01vm
cd /etc/rc0.d; ln -s ../init.d/vm_daemon.sh K01vm
cd /etc/rc6.d; ln -s ../init.d/vm_daemon.sh K01vm
```

Das gezeigte Beispiel paßt für die meisten UNIX-Betriebssystem mit SYSV-ähnlichem `init`-Prozeß. Der Volume Manager ist hierbei aktiv während des "runlevel 3" des Betriebssystems. Die Datei `vm_daemon.sh` ist ein Shell-Skript, das eines der Argumente `start` oder `stop` benötigt. Je nach Art des gewählten Arguments ist sie in der Lage den DM zu starten oder zu stoppen.

Auf anderen UNIX-Betriebssystemen muß eine geeignete Stelle in den Dateien `/etc/rc.local` bzw. `/etc/rc` gefunden werden¹¹. In eine dieser Dateien trägt man an geeigneter Stelle (meist am Ende der Datei) folgende Zeilen ein:

```
# Starten des Datentransportservices
if [ -f /etc/vm_daemon.sh ]; then
    /etc/vm_daemon.sh start
fi
```

Schließlich muß noch das Skript `vm_daemon.sh` in das Verzeichnis `/etc` kopiert werden.

Auf ähnliche Weise läßt sich auch ein Device Manager (`dm`) für einen dauerhaften Programmlauf konfigurieren (siehe auch [DM, Kapitel 4.2]). Hierbei kommt das Skript `dm_daemon.sh` zum Einsatz.

6.2.3.2 Datenbank

Der Volume Manager verwendet eine mSQL-Datenbank. Zur Installation und Konfiguration des Datenbank-Servers `msqld` muß an dieser Stelle auf die mSQL-Dokumentation verwiesen werden. Die folgenden Installationsschritte setzen entsprechende Zugriffsrechte zur Datenbank voraus (Datei `msql.acl` des Datenbankservers). Es ist empfehlenswert, jedoch nicht

¹⁰Weitere Informationen sind über die UNIX-Manualseiten `inittab(4)` und `init(1m)` zu finden.

¹¹Siehe auch UNIX-Manualseiten `rc(1m)` (HP-UX, AIX, ...) bzw. `rc(8)` (SunOS 4.x, BSD, Ultrix, ...).

notwendig, den Volume Manager auf demselben Rechner zu installieren, auf dem der Datenbankserver `mysqld` läuft.

Zur Generierung einer mSQL-Datenbank dient ein Hilfsprogramm. Im Hauptverzeichnis des *smart*-Softwarepakets läßt sich dieses Hilfsprogramm durch das Kommando "make database" starten. Es handelt sich um ein interaktives, zeilenorientiertes Konfigurationsskript. Es verlangt u. a. die Angabe eines Datenbanknamens zur Verwendung für *smart*.

Nach Abschluß der Konfiguration generiert das Skript alle zum Betrieb notwendigen Tabellen der *smart*-Datenbank.

Für einen automatischen Programmstart und "shutdown" des mSQL-Servers steht ein entsprechendes Skript zur Verfügung `mysql_daemon.sh`, welches wie das Skript `vm_daemon.sh` aus Abschnitt 6.2.3.1 verwendbar ist.

Das Datenbankschema ist im Perlmodul "db.pm" des Volume Managers beschrieben. Über dieses Modul kann im begrenztem Umfang eine Anpassung (finetuning) der Datenbank erfolgen.

Bei mSQL handelt es sich um ein einfaches Datenbanksystem mit kleinem Funktionsumfang. So fehlen z. B. Vorkehrungen zur Sicherung der Datenbankkonsistenz nach Systemabstürzen (Transaktionslogbuch). *smart* enthält deshalb ein kleines Hilfsprogramm `mysql-cron-dump`, das den Inhalt einer mSQL-Datenbank kopiert und in entsprechenden Archivdateien ablegen kann. Dieses Hilfsprogramm sollte auf dem Datenbankserver über den Betriebssystemdienst `cron` regelmäßig (z. B. mehrmals täglich) benutzt werden¹².

6.2.3.3 Benutzerschnittstelle

Die VM-Benutzerschnittstelle besteht aus den im Abschnitt 4.3 beschriebenen Hypertextdokumenten. Im *smart*-Softwarepaket sind sie in folgenden Unterverzeichnisse zu finden:

`www/htdocs` – statische Hypertextdokumente;

`www/cgi-bin` – dynamische Hypertextdokumente in Form von CGI-Skripten (Perl-Skripte).

Die Vorbereitung und Installation der Benutzerschnittstelle erfolgt mit den in Abschnitt 6.1.2 dargestellten `configure`- und `make`-Kommandos.

Eine Kontrolle der Benutzer gegen unbefugten Zugriff (Autorisierung) erfolgt einerseits durch die CGI-Skripte. Sie vergleichen die vom WWW-Ser-

¹²Siehe UNIX-Manualseiten zu `crontab(1m)` und `crond(8)`.

ver bei jedem Zugriff eines Benutzers übermittelten Daten (Benutzername und Gruppe) mit den in der Datenbank abgelegten Informationen.

Den Namen eines Benutzers ermittelt der WWW-Server anhand der vom Benutzer (Browser) bei jedem Zugriff mitgeteilten Autorsierungsdaten (Benutzername und Passwort). Zu deren Kontrolle benötigt der WWW-Server Zugang (readonly) zu den Benutzerdaten der Datenbank¹³. Als Beispiel wird die Konfiguration eines Apache-WWW-Server Version 1.1.1 mit Autorisierungsmodul `mysql_auth_module` gezeigt. In der Datei `access.conf` des WWW-Servers sind folgende Zeilen anzufügen:

```
<Directory /home/backup/smart/www>
  Auth_MSQLhost      localhost
  Auth_MSQLdatabase  smart
  Auth_MSQLpwd_table http_auth_user
  Auth_MSQLuid_field user
  Auth_MSQLpwd_field passwd
  Auth_MSQLgrp_table http_auth_group
  Auth_MSQLgrp_field group
  Auth_MSQL_nopasswd off
  Auth_MSQL_Authorative on
  Auth_MSQL_EncryptedPasswords on
  <Limit GET POST>
    require valid-user
  </Limit>
</Directory>
```

Dabei sind einige Angaben anzupassen:

- `Directory` enthält den zur Konfiguration verwendeten Prefix (hier: `/home/backup/smart`).
- `Auth_MSQLhost` muß durch den Rechnernamen ersetzt werden, auf dem die mSQL-Datenbank läuft.
- `Auth_MSQLdatabase` bezeichnet den in Abschnitt 6.2.3.2 gewählten Namen der *smart*-Datenbank.

Die folgenden Zeilen gehören in die Datei `srm.conf` des WWW-Servers und legen Pfadnamen und Zugriffsart für statische und dynamische Hypertextdokumente fest:

```
ScriptAlias /smart/cgi-bin/ /home/backup/smart/www/cgi-bin/
Alias       /smart/         /home/backup/smart/www/htdocs/
```

¹³Konfiguration der Datenbank: siehe mSQL-Dokumentation.

Damit wird die Leitseite der *smart*-Hypertextdokumente als URL bestimmt:
<http://wwwserver/smart/index.html>.

Die Benutzerschnittstelle lässt sich leicht für einen anderen WWW-Server und dessen Zugangskontrolle anpassen. Informationen darüber sind in den Quelltexten von *smart* im CGI-Skript `user-admin.pl` enthalten.

Die Generierung einer neuen *smart*-Datenbank (Abschnitt 6.2.3.2) mit "make database" verwendet einige Anfangswerte. Dazu gehört ein Benutzereintrag mit Passwort als *smart*-Operator, bestehend aus Informationen über den Benutzer, der die Installation der Software durchführt. Das verwendete Passwort wurde dabei aus `/etc/passwd` kopiert oder, falls dies nicht möglich war, ein neues generiert und dem Benutzer mitgeteilt.

Die nachfolgende Konfiguration von *smart* erfolgt über die *smart*-Benutzerschnittstelle.

6.3 *smart*-Konfiguration

Dieser Abschnitt beschreibt wichtige Aspekte der *smart*-Konfiguration.

Einige Stellen der Konfiguration verwenden den Begriff "laufende Nummer". Diese Nummern identifizieren einzelne Einträge in der Datenbank. Jeder Ersteintrag erhält eine neue laufende Nummer zugewiesen. Diese Nummern sind eindeutig und werden nach der Löschung eines Eintrags nicht wiederverwendet. Der Wert einer laufenden Nummer hat keine Bedeutung für die Funktion des Volume Managers.

6.3.1 Benutzerverwaltung

Benutzer, die auf Funktionen von *smart* zugreifen wollen, müssen als Administrator in der Benutzerverwaltung eingetragen sein. Dieser Eintrag besteht, neben einer Benutzer-ID und Passwort, aus Angaben zu Name, Vorname des Administrators und der Gruppen, denen er angehört. Außerdem können weitere Angaben eingetragen werden, die es anderen Benutzer ermöglicht, einen bestimmten Administrator anzusprechen: Telefonnummer, Arbeitsplatz, Abteilung, Raum sowie seine Email-Adresse. Diese Informationen werden von einigen Hypertextdokumenten verwendet, um z. B. bei Problemen schnell den für einen Client zuständigen Administrator zu finden.

Jeder Benutzer darf seinen eigenen Eintrag in der Benutzerverwaltung verändern. Davon ausgenommen sind Gruppenberechtigungen, die nur von Administratoren der privilegierten Gruppe `operator` verändert werden

dürfen. Operatoren dürfen zusätzlich Einträge aller Benutzer verändern, Einträge für neue Benutzer erstellen und Benutzer löschen.

6.3.2 Zeitrahmen

Bei der Konfiguration werden Kurzbezeichnungen für Zeitrahmen vereinbart. Solche Kurzbezeichnungen sollen die ihnen zugeordneten Zeitrahmen treffend beschreiben (z. B. "Nachts", "Morgens" oder "Wochenende"). Jeder vereinbarten Kurzbezeichnung lassen sich mehrere Zeitrahmen als Liste zuordnen. Diese Kurzbezeichnungen werden beim Eintrag einer Datensicherung verwendet. Es lassen sich damit einzelne Zeitrahmen global für alle Datensicherungen verändern, die zu der betreffenden Kurzbezeichnung gehören. Werden mehrere Zeitrahmen in einer Liste zusammengefaßt, dürfen sie sich gegenseitig überlappen.

Es gibt zwei Arten von Zeitrahmen: tägliche und wöchentliche. Ein täglicher Zeitrahmen öffnet jeden Tag Zeitfenster, während der Datensicherungen durchgeführt werden dürfen. Ein wöchentlicher Zeitrahmen ist dagegen an einen bestimmten Wochentag gebunden.

Die Definition eines Zeitrahmens für eine tägliche bzw. wöchentliche Wiederholung ist von der Angabe eines Intervalls von Datensicherungen zu unterscheiden. Ein Intervall wird zur Berechnung von Sicherungszeitpunkten basierend auf Zeitrahmen verwendet.

In der folgenden Darstellung der Spezifikation von Zeitrahmen steht "WWW" für die ersten zwei oder drei Buchstaben eines Wochentags, "HH" für eine Stundenangabe und "MM" für Minuten. Zeitrahmen können in den folgenden Formaten angegeben werden.

- "HH:MM-HH:MM": Zeitraum mit je einer Uhrzeit als Start und Ende, Beispiele:
 - "04:00-07:30": 4⁰⁰ Uhr bis 7³⁰ Uhr jeden Morgen.
 - "20:00-07:30": 20⁰⁰ Uhr Abends bis 7³⁰ am nächsten Morgen.
 - "00:00-23:59": Zeitrahmen, der eine Datensicherung zu jeder Zeit erlaubt. Sicherungszeitpunkte und Intervalle werden dabei basierend auf Mitternacht berechnet, d. h. Datensicherungen mit einem täglichen Intervall werden vom Volume Manager nach Mitternacht gestartet.
 - "20:00-19:59": Wie zuvor, jedoch erfolgt die Berechnung von Sicherungszeitpunkte und Intervalle bezogen auf 20 Uhr.
- "WWW, HH:MM-HH:MM": Zeitraum von maximal 24 Stunden beginnend an einem bestimmten Wochentag, Beispiele:

- "Mon, 04:00-07:30": 4⁰⁰ Uhr bis 7³⁰ Uhr jeden Montag Morgen.
- "Wed, 20:00-07:30": Mittwoch Abend, 20⁰⁰ Uhr bis 7³⁰ Uhr am nächsten Morgen.
- "WWW, HH:MM-WWW, HH:MM": Zeitraum von mehr als 24 Stunden, Beispiel:
 - "Fri, 23:00-Mon, 06:15": Freitag Abend bis Montag Morgen (Wochenende).

Alle Zeitangaben werden vom Volume Manager nach Ortszeit interpretiert (Sommer- bzw. Winterzeit).

Die *smart*-Benutzerschnittstelle akzeptiert bei der Eingabe von Zeitrahmen Variationen des oben angegebenen Formats. So werden z. B. Groß- und Kleinschreibung nicht unterschieden. Als Wochentage sind deutsche und englische Abkürzungen erlaubt. Führende Nullen bei Zeitangaben dürfen entfallen. Die CGI-Skripte korrigieren die Schreibweise eines Zeitrahmens und melden eine normierte Schreibweise nach erfolgtem Eintrag als Ergebnis.

Ein Zeitrahmen bestimmt nur die Zeitpunkte, zu denen der Volume Manager eine Datensicherung starten darf. Er kontrolliert nicht deren Ende, so daß eine bereits laufende Sicherung weit über das Ende eines Zeitrahmens hinaus andauern kann. Bei der Festlegung von Zeitrahmen sollte dieser Umstand entsprechend berücksichtigt werden.

6.3.3 Access-Control-Listen

Access-Control-Listen (ACL) bestimmen die Zugriffsrechte auf die gesicherten Daten (Volumes). Eine ACL besteht aus einer Anzahl von Gruppennamen. Bei der Konfiguration einer Datensicherung kann jedem Eintrag eine ACL zugeordnet werden. Wird einer solchen Liste eine Gruppe hinzugefügt oder daraus entfernt, so ändern sich die Zugriffsrechte aller mit dieser Liste verknüpften Datensicherungen.

6.3.4 Cache

Ein Cache wird bei der Datensicherung und Restaurierung als Zwischenspeicher genutzt. Eine Cache-Datei muß als *File-Resource* eines Device Managers angegeben werden (Format: siehe Anhang A.1.2). Die Cache-Dateien dürfen auf beliebigen Clients liegen.

Bei der Spezifikation einer Cache-Datei sind anzugeben:

- Die maximale Größe der Datei. Es ist empfehlenswert Cache-Dateien verschiedener Größe anzulegen: z. B. drei Files mit 10 MByte und sechs Files mit 100 MByte. Die Größe einer Cache-Datei wird in Byte angegeben. Durch Anhängen eines Buchstabens läßt sich die Größe in KByte "k" oder MByte "m" ausdrücken. Bei Angabe mehrerer Größen wird deren Summe verwendet. Beispiel: Eine Größe von 2.5 MByte kann angegeben werden als "2621440", "2560k" oder "2m512k".
- Eine Priorität beschränkt die Nutzung eines Cache auf Datensicherungen mit gleicher oder höherer Priorität. Die Priorität des Cache kann dabei als Dezimalzahl angegeben werden. Eine Cache-Priorität von z. B. 4.5 kann vom Volume Manager für eine Datensicherung eingesetzt werden, die eine Anfangspriorität von 4 hatte und bereits für die Dauer eines halben Intervalls verzögert wurde (Verzögerungsfaktor 0.5)¹⁴.
- Eine Liste von Gruppen. Eine Datensicherung, für die ein Cache genutzt wird, muß mindestens einer der aufgeführten Gruppen angehören. Die Gruppenliste darf leer sein. In diesem Fall, darf der Volume Manager die Cache-Datei für jede Sicherung einsetzen.

Durch die Größenangaben der Cache-Dateien läßt sich der maximale Plattenspeicherbedarf für Datensicherungen berechnen. Diesen Platz belegt der Volume Manager nur während der Durchführung von Datensicherungen (z. B. Nachts). Es kann daher sinnvoll sein, diesen Platz außerhalb der konfigurierten Zeitrahmen anderweitig zu verwenden (z. B. tagsüber). Falls der Volume Manager nicht die angegebenen Kapazitäten nutzen kann, weil der Plattenplatz noch anderweitig belegt ist, reduziert der Volume Manager seinen Platzbedarf vorübergehend, d. h. er nutzt den konfigurierten Cache nicht aus. Datensicherungen werden dadurch nicht beeinträchtigt¹⁵. Sie werden also ohne Fehler fortgesetzt.

Für jedes eingesetzte Bandlaufwerk sollten mindestens zwei Cache-Dateien bereitgestellt werden. Auch sollten einzelne Cache-Dateien nicht zu klein gewählt werden, da der Volume Manager bzw. die Bandlaufwerke sonst zu viel Zeit und Bandkapazität für Verwaltungsaufgaben benötigen (Schreiben von End-Of-File-Marken und Filelabel, Bandpositionierung, etc.).

¹⁴Die Berechnung der Sicherungspriorität ist in Abschnitt 4.2.6 beschrieben.

¹⁵Eine indirekte Beeinträchtigung existiert dennoch: Der Datendurchsatz und damit die Sicherungsgeschwindigkeit wird geringer.

6.3.5 Bandlaufwerke

Bandlaufwerke zur Datensicherung müssen als *Device-Resource* eines Device Managers angegeben werden (Format: siehe Anhang A.1.2). Die Bandlaufwerke dürfen an beliebigen Clients angeschlossen sein. Bandlaufwerke mit Bandwechselroboter können auf eine bestimmte Anzahl Bandmedien aus einem Magazin automatisch zugreifen.

Bei der Spezifikation eines Bandlaufwerks darf eine Priorität und eine Liste von Gruppen (ACL) angegeben werden. Priorität und Gruppen haben dieselbe Bedeutung wie bei der Spezifikation einer Cache-Datei. Sie beschränken die Nutzung eines Bandlaufwerks.

6.3.6 Datensicherungen

Das Hypertext-Formular zur Anmeldung und Änderung von zu sichernden Datenbestände verlangt die Angabe einer Anzahl von Argumenten. Diese Argumente beschreiben, welche Daten, wann und wie gesichert werden und bilden damit die Grundlage der Planung des Volume Managers.

“**Host**”:
Rechnername des zuständigen Device Managers, über den die Datensicherung gestartet werden soll.

“**Filesystem, Pfad, Partition**”:
identifiziert die zu sichernden Daten auf “Host”. Diese Angabe wertet nur das angegebene Datensicherungsskript aus. Die Werte von “Host” und “Filesystem, Pfad, Partition” sind sogenannte Primärschlüssel der Datenbank, d. h. jedes Wertepaar kann deshalb nur einmal eingetragen werden.

“**Zeitraumen**”:
Kurzbezeichnung einer Liste von Zeitraumen, zu denen eine Datensicherung gestartet werden darf (siehe Abschnitt 6.3.2).

“**Intervall**”:
Zeit zwischen zwei aufeinanderfolgenden Datensicherungen (siehe Abschnitt 4.2.3). Als Maßeinheiten lassen sich Minuten “m”, Stunden “h”, Tage “d” oder Wochen “w” angeben. Mehrere Angaben werden summiert. Beispiele:

- “1440”, “1440m”, “24h”, “1d”: verschiedene Schreibweisen für ein Intervall von einem Tag (24 Stunden).
- “3d12h”: 3 Tage und 12 Stunden = 84 Stunden.

Die Planung eines Sicherungszeitpunkts führt der Volume Manager anhand der Angaben “Zeitraumen” und “Intervall” durch (siehe Abschnitt 4.2.5).

- “**Anfangspriorität**“: gibt einer Sicherung einen Rang im Bereich von 0 bis 9 (siehe Abschnitt 4.2.6).
- “**verbinden mit**“: verweist auf die “laufende Nummer” eines anderen Planungseintrags (siehe Abschnitt 4.2.7).
- “**Zugriffsrecht**“: Liste von Gruppen (ACL) – bestimmt, wer auf die gesicherten Daten zugreifen darf (siehe Abschnitt 6.3.3). Während einer Datensicherung kann die Nutzung von Cache-Dateien und Bandlaufwerken über diese Liste eingeschränkt sein (siehe Abschnitt 4.2.8).
- “**Aufbewahrungsfrist**“: klassifiziert eine Datensicherung bezüglich einer Archivierungsfrist (siehe Abschnitt 4.2.9).
- “**Sicherungsskript, -Programm**“: ein Datensicherungsskript auf Rechner “*Host*“, das die zu sichernden Daten (“*Filesystem, Pfad, Partition*“) als Datencontainer verpackt (siehe Abschnitt 4.2.10). Die Angabe darf sogenannte Meta-Argumente enthalten. Diese Meta-Argumente werden vom Volume Manager durch jeweils aktuelle Werte ersetzt.

Argument	wird ersetzt durch
%H	“ <i>Host</i> “
%F	“ <i>Filesystem, Pfad, Partition</i> “
%P	“ <i>Priorität</i> “
%I	“ <i>Intervall</i> “
%T	“ <i>Zeitraumen</i> “ (timeframe)
%E	“ <i>Aufbewahrungsfrist</i> “ (expire)
%A	“ <i>Zugriffsrecht</i> “ (ACL)
%V	Volume-ID des Datencontainers
%L	Liste aller bekannter Dump-IDs/Volume-IDs seit der letzten Gesamtsicherung einschließlich
%%	% (Prozentzeichen)

Bei den für “%L” als Textersatz genannten Liste von Dump-IDs (siehe Seite 42) handelt es sich um eine durch Kommas getrennte Liste von Identifier, auf die eine neue inkrementelle Sicherung aufbauen darf. Falls ein Datensicherungsskript keine eigenen Identifier verwendet, besteht diese Liste aus Volume-IDs. Eine leere Liste bedeutet, daß das Sicherungsskript eine Gesamtsicherung erstellen muß. Über diese Angabe kann der Volume Manager die Wahl eines neuen dump level einschränken.

Beispiel: Eine Skript-Angabe von

```
do-backup -l %L %F
```

mit *“Host”* `inf` und *“Filesystem”* `/usr`, sowie der auf Seite 42 gezeigten Dump-IDs, führt zu folgendem DM-Systemkommando (vergleiche Anhang A.1.2):

```
inf:S'do-backup -l 19961118.223757,19961122.224255 /usr'
```

Eine Angabe von *“%L”* ist empfehlenswert, da ohne dieses Meta-Argument der Volume Manager keine Möglichkeit hat, gezielt eine Gesamtsicherung zu erstellen.

Für eine weitergehende Beschreibung verfügbarer Datensicherungsskripte, deren Kommandooptionen, -Argumente und Syntax, sowie Zweck und Verwendung der Meta-Argumente, muß an dieser Stelle auf die im *smart*-Softwarepaket enthaltenen Informationen verwiesen werden.

Kapitel 7

Ausblick

smart stellt für die Aufgaben einer zentralen Datensicherung keine endgültige Systemlösung dar. Vielmehr kann die *smart*-Software als Basis für weitere Entwicklungen betrachtet werden. Sie bietet viele Möglichkeiten zur Anpassung von Methoden und Funktionen an veränderte Anforderungen.

Device Manager — Die Software des Device Managers hat sich während der Entwicklung von *smart* als zuverlässig erwiesen. Erweiterungen im Funktionsumfang sind u. a. im Bereich der Ansteuerung von Bandlaufwerken und Bandwechselsystemen möglich.

Volume Manager — Durch den Einsatz von Perl als Implementierungssprache ist der Volume Manager als Kernstück der *smart*-Software auf vielen UNIX-Rechnern unverändert einsetzbar.

Seine modulare Struktur und die Steuerung einzelner Funktionen über Tabellen (endliche Automaten), erlauben eine rasche Anpassung der Software an neue Aufgaben und Anforderungen.

Datensicherungsprogramme — Die Funktionen von Volume und Device Manager sind von der eingesetzten Datensicherungsmethoden unabhängig. Neue Methoden und Sicherungsprogramme können leicht durch eine Anpassung der Sicherungsskripte integriert werden.

Benutzerschnittstelle — Die Realisierung der *smart*-Benutzerschnittstelle als Hypertextdokumente bietet viel Raum für Erweiterungen.

Desweiteren sind verschiedene andere Benutzerschnittstellen realisierbar, die ohne einen WWW-Server funktionieren. Als Beispiel soll ein Shellkommando `vmcp` genannt werden, das ähnlich dem Kommando `dmcp`, Funktionen des Volume Managers direkt zum Kopieren von Volumes (Datencontainern) nutzen kann.

`xdmui` demonstriert als X11-Applikation einen anderen Weg zu Realisierung von Benutzerschnittstellen. Unter Einsatz des Perl-Moduls `Sx` oder des mächtigeren Perl-Tk-Moduls sind vielfältige und leistungsfähige graphische Benutzerschnittstellen möglich.

Datenbank — Für *smart* wird z. Z. das Datenbanksystem `mSQL` der Version 1.0.16 eingesetzt. Die Software der Nachfolgeversion 2.0 befindet sich z. Z. im sogenannten beta-Stadium¹. Mit ihrer Verfügbarkeit ist im ersten Quartal 1997 zu rechnen. Da es sich bei der neuen Version um ein, gegenüber der alten Version, stark verbessertes Datenbanksystem mit erweitertem Funktionsumfang handelt, erscheint es sinnvoll, *smart* entsprechend anzupassen.

Alternativ bestehen über verschiedene Perl-Module Möglichkeiten zur Anbindung an andere Datenbanksysteme (`Informix`, `Ingres`, `Oracle`, `Postgres95`, u. a.).

Anwendungsgebiet — Der Einsatz von *smart* muß nicht auf Aufgaben einer zentralen Datensicherung beschränkt bleiben. Mit einer entsprechenden Erweiterung der Software läßt sich ein "*Hierarchical Storage File System*" (Virtuelles Filesystem großer Kapazität auf Bandmedien) realisieren. Ein solches Filesystem wird in [HSM] beschrieben.

Dabei handelt es sich um ein großes Filesystem. Die auf einer Festplatte gespeicherten Daten werden bei Bedarf auf ein Bandmedium ausgelagert, "migriert". Der dadurch freiwerdende Plattenplatz kann von den Benutzern des virtuellen Filesystems für weitere Daten genutzt werden. Sobald ein Benutzer eine migrierte Datei benutzen möchte, kopiert das virtuelle Filesysteminterface die gewünschte Datei automatisch von Band zurück auf die Festplatte. Den Benutzern steht damit ein Filesystem zur Verfügung, dessen nutzbare Kapazität von der Größe des Bandsystems abhängt und nicht von der verwendeten Festplatte.

Die Funktionen eines migrierenden Filesystems ähneln dem einer zentralen Datensicherung: Datenbestände werden auf Bandmedien kopiert und später, nach deren Löschung, wieder restauriert. Damit *smart* für ein solches virtuelles Filesystem eingesetzt werden kann, ist ein entsprechendes Interface (Filesystemimplementierung, z. B. auf der Basis von NFS) zwischen dem Volume Manager und dem Betriebssystem notwendig.

smart ist also eine leistungsfähige Software mit vielfältigen Erweiterungsmöglichkeiten. Sie wird vom Autor auch nach Abschluß dieser Diplomarbeit weiterentwickelt und gepflegt.

¹Stand: Dezember 1996

Anhang A

Device Manager

Die folgenden Abschnitte beschreiben Syntax und Semantik von Kommandos und Objektnamen des Device Managers. Texte und Diagramme wurden dabei zum Teil aus [DM] übernommen und ergänzt, soweit es zu Verständnis und Anwendung von *smart* und der DM-Benutzerinterfaces nötig ist. Funktionale Erweiterungen des Device Managers, die zusammen mit *smart* entwickelt wurden, sind als Fußnoten vermerkt.

Device Manager wird über Kommandos gesteuert. Kommandos werden z. B. von Benutzern über entsprechende Interfaces oder direkt über Telnet (siehe [RFC 854]) an einen DM gerichtet. Ebenso können andere Programme, wie z. B. der Volume Manager von *smart*, Kommandos an einen DM geben. Aus der Sicht des DM sind diese Programme ebenfalls Benutzer der DM-Dienste. Deshalb werden in diesem Teil des Anhangs alle Instanzen, die Kommandos an einen DM geben, als Benutzer bezeichnet.

Die meisten Kommandos enthalten als Argument ein oder zwei Objektnamen, die zur Ausführung der Kommandos verwendet werden. Es sind drei verschiedene Objekttypen zu unterscheiden:

Resource: Als *Resource* werden alle Instanzen eines Rechners bezeichnet, die Daten speichern, aufnehmen oder abgeben können. Dazu gehören neben einfachen Dateien auch Bandlaufwerke und Programme, die über eine UNIX-Pipe Daten vom DM übernehmen oder an ihn abgeben.

Alias: Jede *Resource* muß, bevor der DM damit arbeiten kann, mit dem *assign*-Kommando, das in Abschnitt A.1.3.1 näher erklärt wird, angemeldet werden. Bei dieser Anmeldung entsteht aus einer *Resource* ein symbolischer Name, ein *Alias*. Solange ein *Alias* existiert, kontrolliert der Device Manager den Zugriff auf die mit dem *Alias* assoziierte *Resource*. Während dieser Zeit ist zum Beispiel aus Sicherheitsgrün-

den jeder fremde Zugriff auf ein beim DM angemeldetes Bandmedium gesperrt. Ein *Alias* kann durch das `release`-Kommando wieder freigegeben werden.

Channel: Als *Channel* bezeichnet man ein für den Datentransport vorbereiteten "Datenpfad" zwischen der entsprechenden *Resource* und dem DM. Er wird mit den Kommandos `setinput` und `setoutput` (siehe A.1.3.2) aus einem *Alias* erzeugt. Solange ein *Channel* aktiv ist, kann der dazugehörige *Alias* weder geschlossen noch für einen anderen *Channel* aktiviert werden. Ein *Channel* wird durch das `close`-Kommando geschlossen, wodurch der *Alias* freigegeben wird.

Das eigentliche Transportkommando enthält als Argumente zwei *Channel*: einen, aus dem die Daten vom DM abgeholt, und einen anderen, dem die übertragenen Daten übergeben werden. Diese beiden *Channel* lassen sich als *Input-Channel* und *Output-Channel* bezeichnen.

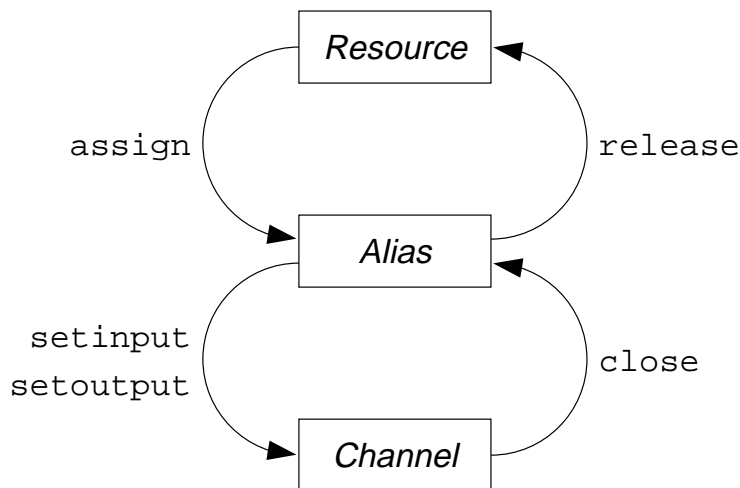


Abbildung A.1: Device Manager – Objekttypen

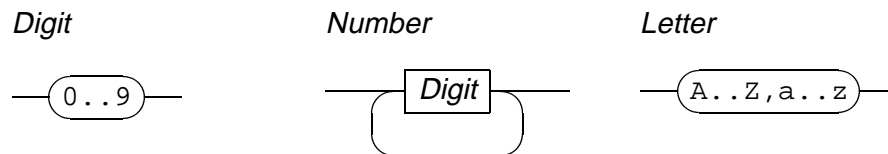
Abbildung A.1 beschreibt die Verbindung einzelner Objekttypen durch Kommandos. Die Beschreibung des Benutzerinterfaces `dmcp` in Abschnitt A.3 enthält Anwendungsbeispiele der hier vorgestellten DM-Kommandos.

A.1 Syntaxdiagramme

Die Syntax soll hier mit Hilfe von Diagrammen verdeutlicht werden. Diese setzen sich aus mehreren Grundelementen zusammen. In einem Syntaxdiagramm heißen Bezeichner in abgerundeten Kästchen dargestellten Zei-

chen Terminal-Symbole¹. Jedes Diagramm beginnt oben links und endet rechts.

A.1.1 Allgemeine Syntaxelemente

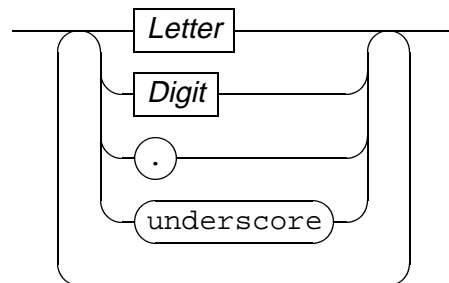


Das erste Diagramm *Digit* beschreibt eine einfache Ziffer.

Eine Zahl (*Number*) besteht aus einer oder mehreren Ziffern. Im zweiten Diagramm wird die Definition *Digit* als Nonterminal-Symbol verwendet. Nonterminals werden *kursiv* in einem eckigen Kästchen dargestellt. Eine beliebige Wiederholung von *Digit* in einer Zeichenkette wird als Schleife gezeichnet.

Letter besteht aus genau einem Buchstaben.

Identifizier

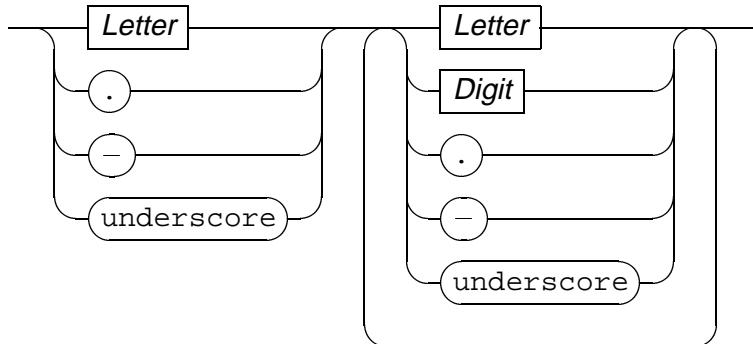
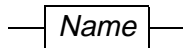
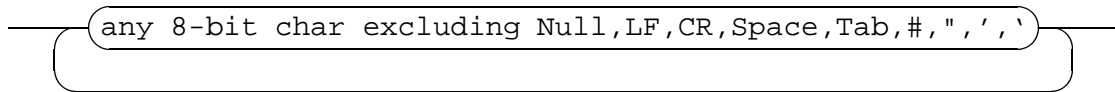


Identifizier ist ein Wort, das sich aus einem oder mehreren Buchstaben, Ziffern, Punkten oder Unterstrich (*underscore*) beliebig zusammensetzt. Im Gegensatz zur sonst gebräuchlichen Definition eines *Identifizier* muß dieser nicht mit einem Buchstaben beginnen.

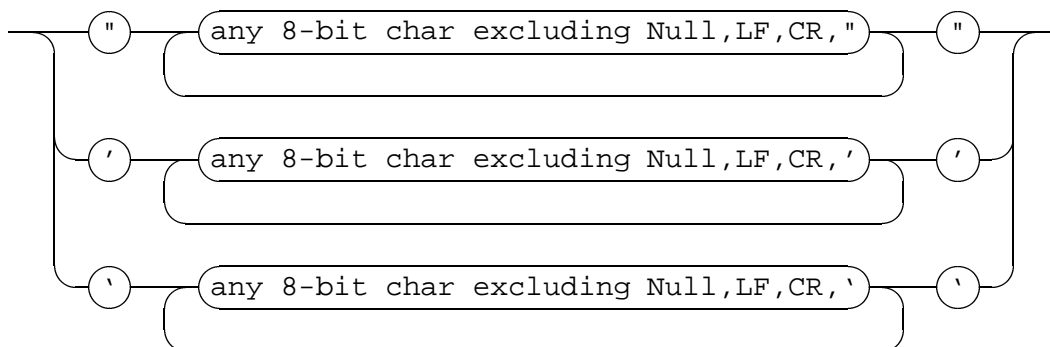
Name ist ein *Identifizier*, der nicht mit einer Ziffer beginnt, dafür aber Bindestriche enthalten darf.

Hostname besitzt dieselbe Syntax wie *Name*. In einigen Diagrammen wird *Hostname* und *Name* unterschieden um hervorzuheben, an welchen Stellen ein Rechnername erwartet wird.

¹Die Begriffe Terminal und Nonterminal werden in [AHO 88] näher erklärt.

Name*Hostname**Pathname*

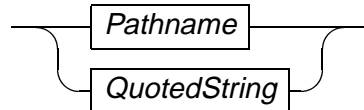
Pathname steht für einen Dateinamen. Dateinamen sind Zeichenketten, in denen einige Zeichen nicht vorkommen dürfen (siehe Diagramm).

QuotedString

QuotedString ist eine Zeichenkette wie *Pathname*. Sie darf jedoch zusätzliche Zeichen als Bestandteil enthalten (z. B. Leerzeichen). Ein *QuotedString* wird in Anführungszeichen geschrieben. Die verwendeten Anführungszei-

chen dürfen in der von ihnen eingeschlossenen Zeichenkette nicht vorkommen.

SysCmd

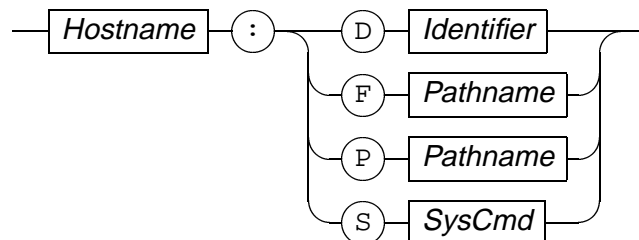


Ein Systemkommando² (*SysCmd*) besteht in der Regel aus einem Programmnamen (Dateiname) und optionalen Argumenten.

A.1.2 DM-Objektnamen

Am Anfang eines Objektnamens steht immer der Name des Rechners, zu dem das betreffende Objekt gehört. Der Objekttyp steht bei einer *Resource*-Angabe als Buchstabe direkt nach dem Rechnernamen und einem Doppelpunkt. Die Bedeutung der nach diesem Buchstaben folgenden Zeichen ist vom Objekttyp abhängig.

Resource



Es werden vier *Resource*-Objekttypen unterschieden:

“**F**” **File**: einfache Datei;

“**P**” **Pipe**: Schnittstelle zu anderen Programmen, unter UNIX “named pipe” oder “FIFO” genannt.

“**D**” **Device**: Bandlaufwerke und Bandwechselroboter lassen sich über einen Objektnamen ansprechen. Der verwendete *Identifier* (siehe Diagramm) muß in der Konfigurationsfile `dev.conf` angegeben sein (siehe Anhang A.1.4).

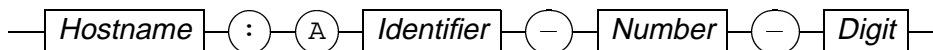
²Systemkommandos werden in [DM] nicht beschrieben; Erweiterung des Device Managers für *smart*.

“s” **Systemkommando**: spezifiziert ein Programm mit optionalen Argumenten, das vom Device Manager direkt ausgeführt werden kann. In Abhängigkeit von der jeweiligen Transportrichtung erzeugt dieses Kommando während eines Transports einen Datencontainer und schreibt diesen auf die Standardausgabe (stdout) oder nimmt einen Datencontainer über Standardinput (stdin) entgegen. Ein Protokoll des Kommandos wird dabei über Standardfehlerausgabe (stderr) vom Device Manager akzeptiert und an den Benutzer weitergeleitet.

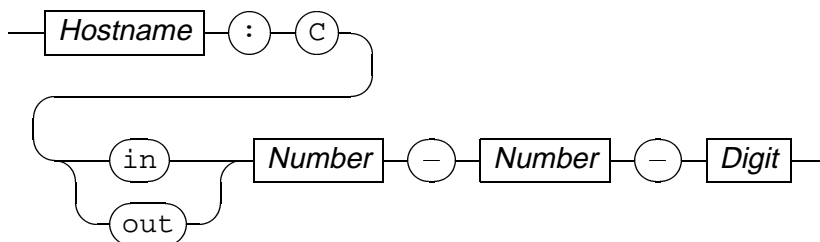
Resource-Angaben werden u. a. bei der Konfiguration des Volume Managers benötigt. Beispiele für *Resources* und deren Bedeutung:

Resource	Bedeutung
inf:F/backup/cache/01	File auf Rechner inf
zdi:P/tmp/dm-out	named-pipe auf Rechner zdi
inf:S'/bin/tar cf - /soft'	Angabe eines Datencontainer-erzeugenden Programms mit Argumenten
inf:Dexa5	Bandlaufwerk mit der Bezeichnung exa5

Alias



Channel



Alias- und *Channel*-Objekte werden unter *smart* intern vom Volume Manager verwendet. Benutzer haben über die DM-Benutzerinterfaces *dmui* und *xdmui* Zugang zu diesen Objekten. *Alias*- und *Channel*-Objektnamen vergibt der Device Manager. Die Bedeutung dieser Namen ist in [DM] ausführlicher behandelt.

A.1.3 DM-Kommandos

Die folgenden Syntaxdiagramme beschreiben die vom Device Manager akzeptierten Kommandos mit Hilfe der bereits vorgestellten syntaktischen Elemente. Einige Teile der folgenden Diagramme sind mit den Buchstaben "D", "F" oder "P" beschriftet. So gekennzeichnete Teile gelten nur für die entsprechenden *Resource*-Typen. Beispiel: Im Diagramm *Command* ist dem Kommando `eject` der Buchstaben "D" zugeordnet. Auf diese Weise markierte syntaktische Elemente sind nur zusammen mit Objekten des angegebenen Typs verwendbar. Das `eject`-Kommando erwartet demnach als *Resource* ein *Device*-Objekt als Argument.

Eine Kommandozeile besteht aus einem Befehlsword und den dazugehörigen Argumenten. Nonterminals werden in einer Kommandozeile mit Leerzeichen voneinander getrennt. Eine Ausnahme bildet das Gleichheitszeichen. Das Schlüsselwort vor dem "=", das Gleichheitszeichen selbst und der Wert danach müssen ohne Leerzeichen geschrieben werden.

Das Ende einer Kommandozeile bildet ein Zeilenendezeichen: newline (linefeed, LF), carriage return (CR) oder "Telnet end-of-line sequence" (siehe [RFC 854]).

Obwohl die folgenden Syntaxdiagramme eine feste Reihenfolge der Optionen vorschreiben, dürfen sie in beliebiger Reihenfolge nach einer *Channel*-, *Alias*- oder *Resource*-Angabe stehen.

A.1.3.1 `assign`

`assign` meldet ein *Resource*-Objekt beim DM an. Nach erfolgreicher Anmeldung wird ein neuer *Alias* gemeldet.

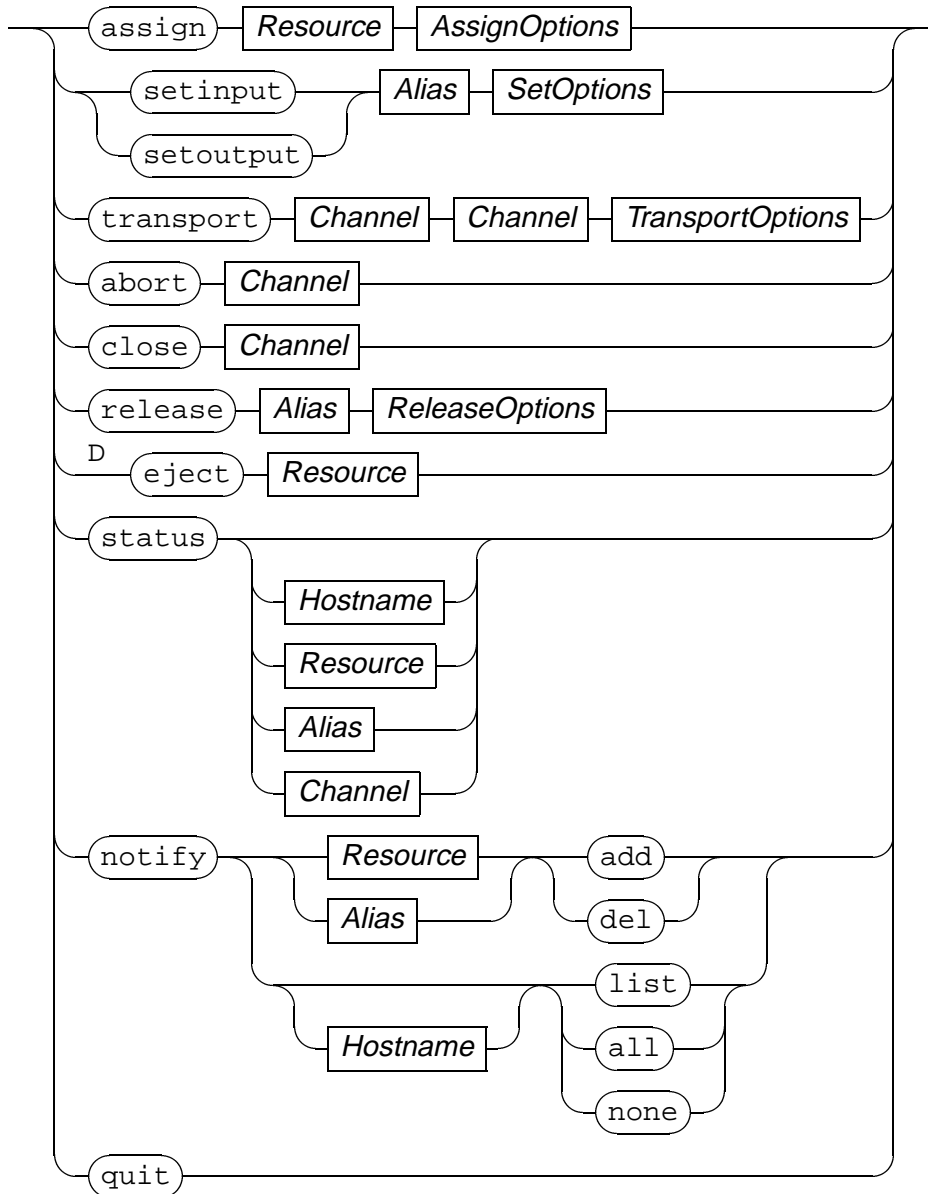
nick: vereinbart einen Kurznamen (nickname). Er wird Bestandteil eines neuen *Alias*.

label: bedeutet dasselbe wie `nick`, wird jedoch für eine *Device-Resource* zur Kontrolle bzw. Erzeugung eines Bandlabels verwendet. Zusammen mit dem Argument "create" wird ein neues Bandlabel auf das Bandmedium geschrieben, wenn eine der folgenden Bedingungen zutrifft:

- Das Band ist leer, d. h. es wurde noch nie beschrieben.
- Das erste File auf dem Band hat die Länge Null.

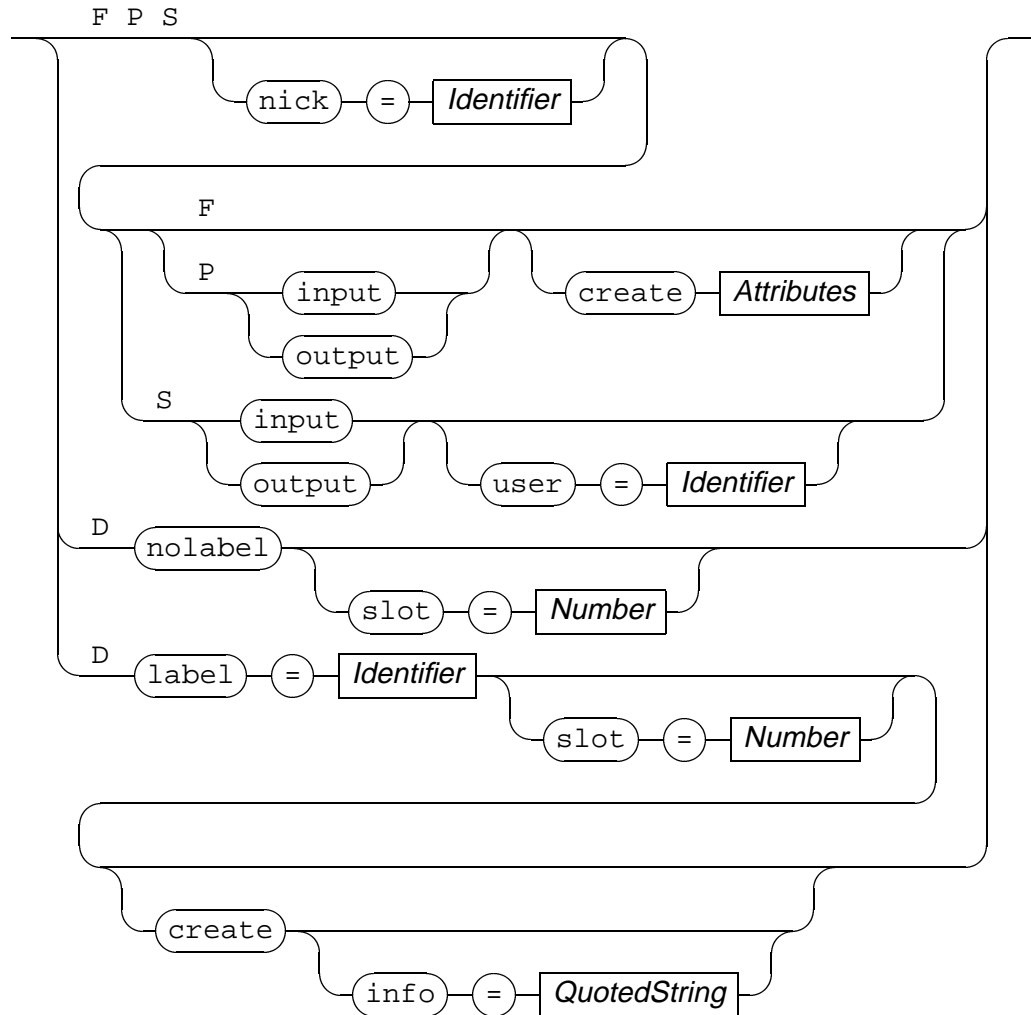
Das Kommando wird abgebrochen, wenn auf einem Band ein anderes als das angegebene Label gefunden wird.

Command



slot: gibt an, aus welchem Schacht der Bandwechselroboter das gewünschte Band holen soll. Befindet sich bereits ein Band im Laufwerk, wird dieses vorher ausgeworfen und in sein ursprüngliches Fach zurückgelegt. Fehlt die `slot`-Angabe, so wird das Band verwendet, das sich im Laufwerk befindet.

AssignOptions



nolabel: Anmeldung eines Bands, das nicht mit dem Device Manager beschrieben wurde und das deshalb kein Bandlabel enthält. Auf diese Bänder kann nur lesend zugegriffen werden. Das Kommando wird abgebrochen, wenn auf einem Band ein Label gefunden wird, obwohl keines erwartet wurde.

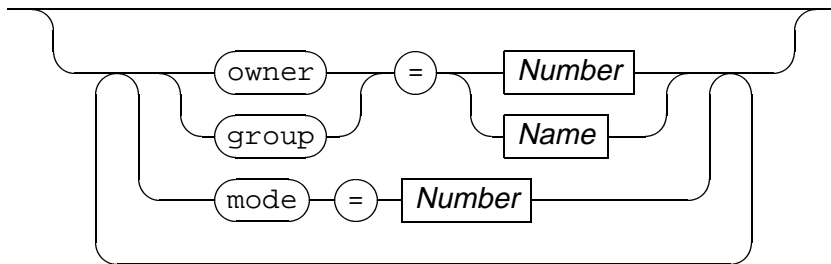
input, output: Bei einer Pipe verlangt das `assign`-Kommando die Angabe, ob der spätere `Alias` zusammen mit `setinput` oder `setoutput` verwendet werden soll.

create: schreibt ein neues Bandlabel (D), öffnet ein neues File (F) oder erzeugt eine neue Pipe (P). Bei File und Pipe ist die Angabe von `owner`, `group` und `mode` möglich.

Attributes owner, group, mode: Spezifikation der Attribute eines neuen Files oder Pipe. Diese Angaben entsprechen den Argumenten der UNIX-Kommandos `chown`, `chgrp` und `chmod`³. `owner` akzeptiert UID-Nummern und Benutzernamen, `group` GID-Nummern und Gruppennamen. `mode` enthält in einer dreistelligen Zahl codiert die Zugriffsrechte für das neue File bzw. die Pipe. Der Wert wird hierbei von der `umask`-Einstellung⁴ des Device Managers beschränkt. `mode` wird grundsätzlich als Oktalzahl interpretiert.

info: nur zusammen mit `create` und *Device-Resource* verwendbar; die angegebene Zeichenkette wird im neuen Bandlabel vermerkt und dient der Identifikation von Aufzeichnungen⁵. Während eines späteren Lesens von Bandinhalten wird von einem `assign`-Kommando dieser Text ausgegeben, sofern der gelesene Bandlabel eine solche Information enthält. Zweckmäßige Inhalte dieses Strings sind Angaben über die gesicherten Datencontainer wie z. B. Ursprung, Sicherungsmethode, etc.

Attributes



DM ergänzt nicht spezifizierte Attribute durch Standardwerte. Neue erzeugte Files bzw. Pipes erhalten die Benutzer- und Gruppenzugehörigkeit, unter der DM gestartet wurde. Für `mode` wird ein geeigneter Wert gewählt.

A.1.3.2 setinput/setoutput

`setinput` öffnet aus einem *Alias* einen *Input-Channel*. Entsprechend öffnet `setoutput` einen *Output-Channel*. Der Benutzer, der einen *Alias* öffnet, ist der Besitzer (`owner`)⁶ des neuen *Channel*. Nur der Besitzer kann

³Siehe UNIX-Manualseite zu `chown(1)`, `chgrp(1)` und `chmod(1)`.

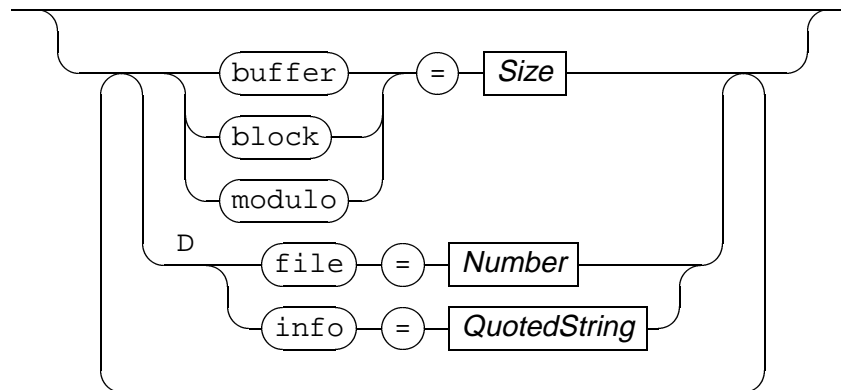
⁴Siehe UNIX-Manualseite zu `umask(1)`.

⁵`info` wird in [DM] nicht beschrieben; Erweiterung des Device Managers für *smart*.

⁶Besitzer eines *Channel* ist ein Benutzer des Datentransportservices. Dagegen ist der Besitzer eines Files bzw. Pipes ein Attribute des Filesystems.

einen *Channel* in anderen Kommandos (`transport`, `abort`, `close`) als Argument verwenden. Davon ausgenommen sind die Kommandos `notify` und `status`, über die jeder Benutzer Informationen eines Objekts abrufen darf.

SetOptions



buffer: Größe des im DM für den Datentransport bereitgestellten Zwischenspeichers in Byte.

block: DM liest bzw. schreibt Daten in Blöcken, deren Größe mit `block` in Byte festgelegt werden kann.

modulo: DM liest bzw. schreibt nur Datenblöcke, deren Länge ein Vielfaches der `modulo`-Angabe sind. Am Ende eines Transports kann es vorkommen, daß trotzdem ein zu kleiner Datenblock gelesen/geschrieben wird, wenn die Gesamtlänge des Datencontainers kein Vielfaches dieses Wertes ist. In diesem Fall erhält der Benutzer eine Warnung. Das Kommando wird abgebrochen, wenn die *Output-Resource* ein *Device* ist, das keine Teilblöcke akzeptiert.

file: nur mit `setinput` und einer *Device-Resource* verwendbar; positioniert das Band auf das angegebene File. Die Zählung der Files⁷ beginnt bei Null. File #0 enthält in der Regel das Bandlabel.

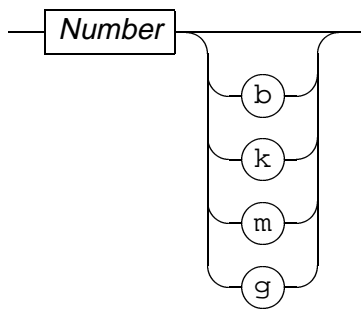
info: nur mit `setoutput` und einer *Device-Resource* verwendbar; die angegebene Zeichenkette wird im Filelabel vermerkt⁸. Ein Filelabel

⁷Bandlaufwerke erlauben nur einen sequentiellen Zugriff auf die Daten eines Mediums. Die einzelnen Aufzeichnungen (Fragmente einer Datensicherung) werden als Files bezeichnet. Sie sind fortlaufend nummeriert.

⁸`info` wird in [DM] nicht beschrieben; Erweiterung des Device Managers für *smart*.

wird vor einem Datencontainer auf Band geschrieben und dient der Identifikation von Aufzeichnungen. Während eines späteren Lesens von Bandinhalten wird vom Kommando `setinput` dieser Text ausgegeben, sofern der jeweilige Label eine solche Information enthält. Zweckmäßige Inhalte dieser Strings sind Angaben über die gesicherten Datencontainer wie z. B. Ursprung, Sicherungsmethode, etc.

Size



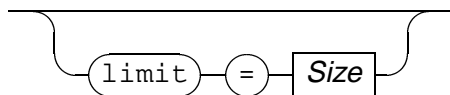
Faktor	Wert
b (Blöcke)	512
k (Kilo)	1024
m (Mega)	1024k
g (Giga)	1024m

Eine Größenangabe (*Size*) besteht aus einer Zahl und einem optionalen Buchstaben als Faktor. Bei `buffer`, `block` und `modulo` der `setinput`/`setoutput`-Kommandos ist die Verwendung des Faktors "g" nicht möglich. Dieser Faktor ist beim Kommando `transport` für die Option `limit` vorgesehen.

A.1.3.3 transport

`transport` verbindet einen *Input-Channel* mit einem *Output-Channel* und beginnt die Datenübertragung. Bei *Channel*, die zu einer *SysCmd-Ressourcen* gehören, werden nun die darin spezifizierten Programme zur Datencontainererzeugung bzw. -Aufnahme gestartet.

TransportOptions



limit: läßt den Transport spätestens nach der Übertragung der angegebenen Anzahl von Bytes abrechen. Danach ist u. a. ein neues `transport`-Kommando möglich. Standardwert ist Null. Null bedeutet unbegrenzt. DM akzeptiert Werte bis zu $10^{18} - 1$.

Nach Ende eines Transports wird für jeden *Channel* die gelesene bzw. geschriebene Anzahl Bytes gemeldet. Bricht ein `transport`-Kommando vorzeitig ab bei Erreichen des angegebenen Limits oder in einer Fehlersituation, können verbleibende *Channel* für ein weiteres `transport`-Kommando verwendet werden. Das Programm einer *SysCmd-Resource* werden in diesem Fall nicht neu gestartet, sondern nur fortgeführt.

A.1.3.4 abort

`abort` bricht einen gerade laufenden Datentransport ab. Danach ist u. a. ein neues `transport`-Kommando möglich.

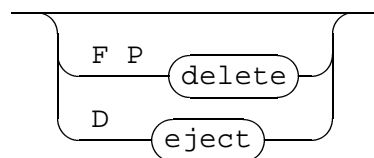
A.1.3.5 close

`close` schließt einen *Channel* und gibt den damit verbundenen *Alias* frei. Das Kommando wird nicht ausgeführt, wenn der angegebene *Channel* Teil eines aktiven `transport`-Kommandos ist oder von einem anderen Benutzer geöffnet wurde. Ein aktiver Transport muß vor einem `close`-Kommando beendet werden (z. B. mit `abort`). Wird die Verbindung zwischen DM und einem Benutzer gelöst, bricht der DM aktive `transport`-Kommandos dieses Benutzers ab und schließt alle *Channel*, die er geöffnet hatte.

A.1.3.6 release

`release` gibt einen *Alias* und die dazugehörige *Resource* frei. Ist der *Resource*-Typ ein File und wurde das File vorher mit `assign/create` erzeugt, dann wird es automatisch gelöscht, falls es leer⁹ ist.

ReleaseOptions



delete: löscht ein zur *Resource* gehörendes File oder Pipe, sofern es vorher mit `assign/create` erzeugt wurde.

⁹File der Länge Null.

eject: veranlaßt das Bandlaufwerk die Kassette auszuwerfen, wenn die Hardware dazu in der Lage ist. Falls die Kassette mit `assign/slot` durch einen Bandwechselroboter eingelegt wurde, legt er die Kassette wieder an ihren ursprünglichen Platz zurück.

Das Kommando wird nicht ausgeführt, wenn mit dem *Alias* ein *Channel* assoziiert ist. Der *Channel* muß vorher mit `close` geschlossen werden.

A.1.3.7 eject

Das `eject`-Kommando arbeitet wie "`release Alias eject`". Jedoch erwartet `eject` als Argument eine *Device-Resource* statt eines *Alias*. Das Kommando wird nicht ausgeführt, wenn die *Resource* angemeldet ist.

A.1.3.8 status

`status` gibt Informationen über das gewünschte Objekt aus. Wenn das Argument ein *Hostname* ist, dann wird eine Liste aller Objekte ausgegeben, die der betreffende DM zur Zeit verwaltet. Die Verwendung von *Hostname* ist hier optional. Wird als Argument ein *Alias* oder eine *Resource* verwendet, so erhält man die damit assoziierten Objekte. Für eine *Device-Resource* werden zusätzliche Angaben über Medienformat und, sofern vorhanden, Anzahl der Slots im Magazin eines Bandwechselsystems gemeldet. *Resources* der Typen *File*, *Pipe* und *SysCmd* erscheinen in einer `status`-Abfragen nur, wenn sie als *Alias* angemeldet sind.

Bei Angabe eines *Channel* werden zusätzliche Informationen über einen eventuell aktiven Transport gegeben. Diese zusätzlichen Informationen können nur vom Besitzer eines *Channel* abgerufen werden.

A.1.3.9 notify

Es kommt bei der Arbeit mit dem Device Manager häufig vor, daß er für einzelne Objekte vorübergehend keine Kommandos akzeptiert. Dies trifft immer dann zu, wenn andere (frühere) Kommandos diese Objekte benutzen und noch nicht freigegeben haben. Mit dem Kommando `notify`¹⁰ ist es leicht möglich, auf die Freigabe eines *Alias*- oder *Resource*-Objekts zu warten. Der Device Manager verwaltet zu diesem Zweck eine Liste, in der alle zu überwachenden Objekte eingetragen sind.

¹⁰`notify` wird in [DM] nicht beschrieben; Erweiterung des Device Managers für *smart*.

add, del: Mit diesen Argumenten werden einzelne *Resource*- oder *Alias*-Objekte in der genannten Liste eingetragen bzw. gelöscht.

list: Ausgabe aller in der Warteliste eingetragener Objekte. Ausgabeformat ist dem des `status`-Kommandos ähnlich.

none: Entfernen aller Einträge aus der Warteliste.

all: Alle bekannten *Resource*- und *Alias*-Objekte werden vom Device Manager automatisch in die Liste aufgenommen. Objekte, die nach einem `notify all`-Kommando erzeugt werden, erhalten automatisch einen Eintrag in der Warteliste.

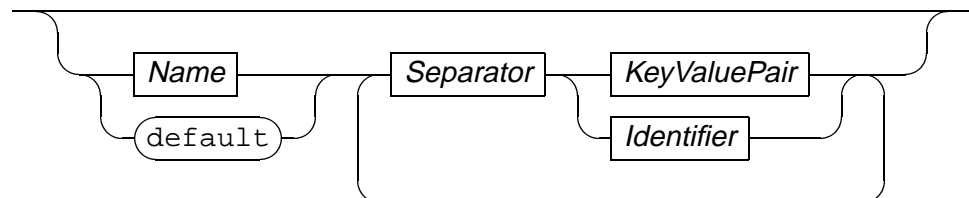
A.1.3.10 quit

`quit` beendet die Verbindung zwischen DM und Benutzer. Aktive `transport`-Kommandos des Benutzers werden abgebrochen und alle *Channel*, die er geöffnet hatte, geschlossen. Dasselbe geschieht auch dann, wenn die Verbindung zwischen DM und Benutzer aus einem anderen Grund gelöst wird.

A.1.4 Syntax Konfigurationsfiles

Konfigurationsfiles bestimmen das Verhalten eines Device Managers. Device Manager versucht beim Programmstart zunächst einen Filenamen mit angehängtem Rechnernamen zu öffnen (z. B. `dm.conf.inf` auf dem Rechner `inf`). Existiert dieses File nicht, wird der ursprüngliche Filenamen verwendet (`dm.conf`). Der Aufbau von Konfigurationsfiles wird mit Syntaxdiagrammen beschrieben.

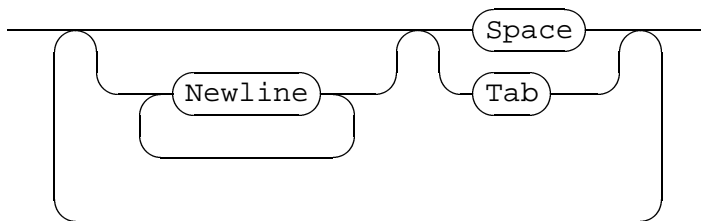
configuration



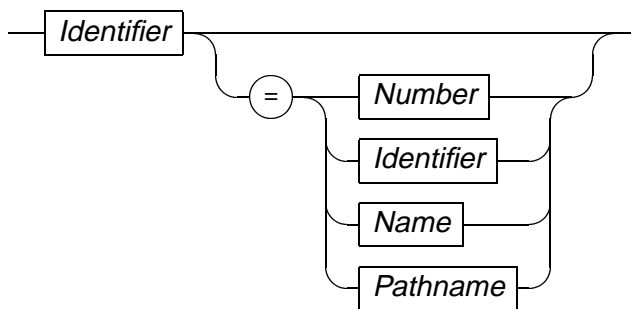
Konfigurationsfiles enthalten eine Anzahl von Einträgen. Diagramm *configuration* zeigt eine allgemeine Darstellung eines Eintrags, bestehend aus einem Namen oder dem Schlüsselwort `default` am Anfang einer Zeile.

Danach können mehrere durch Zwischenraum (*Separator*) getrennte Argumente folgen. Argumente eines Eintrags bestehen aus einem *KeyValuePair* oder einem *Identifier*. Ein Eintrag darf sich über mehrere Zeilen erstrecken,

Separator



KeyValuePair



wobei die zweite und jede weitere Zeile unbedingt mit einem Leerraum einzurücken ist (Diagramm *Separator*). Leerzeilen und Zeilen, die mit dem Kommentarzeichen “#” beginnen, werden ignoriert¹¹. [DM] enthält einige Konfigurationsbeispiele für die nachfolgenden Definitionen.

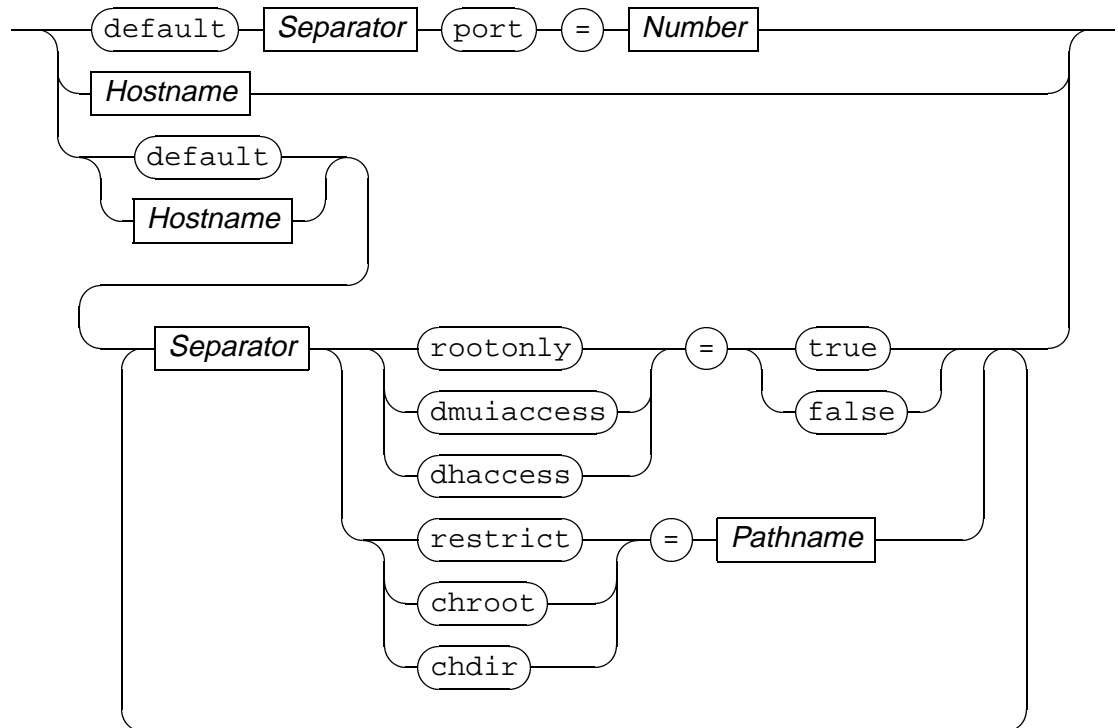
A.1.4.1 dm.conf

Eine Konfigurationsdatei `dm.conf` hat zwei Aufgaben. Zum einen legt sie eine vom Standard abweichende Portnummer fest, über die ein DM erreichbar ist. Zum anderen bestimmt sie, von welchen Rechnern Verbindungen zum DM akzeptiert werden und welche Beschränkungen für die jeweilige Verbindung gelten¹².

¹¹Darstellung von Kommentaren ist im Diagrammen *Separator* nicht enthalten.

¹²Der Schutz gegen unbefugten Zugriffs wurde während der Entwicklung von *smart* um die Optionen `dmuiaccess`, `dhaccess` und `chroot` erweitert.

dm.conf



restrict: legt den Pfad fest, der für eine *File* oder *Pipe-Resource* verwendet werden kann. Die Angabe hat die Form “restrict=PREFIX”, wobei der Pfadname, den ein Benutzer angibt, mit *PREFIX* beginnen muß. Relativen Pfadnamen werden vom Device Manager automatisch durch *PREFIX* ergänzt. Pfadnamen, die zwei Punkte (..) als Komponente enthalten, werden bei Verwendung von `restrict` durch den DM aus Sicherheitsgründen abgelehnt.

rootonly: Für jede Verbindung eines Benutzers zum DM ist deren Ausgangspunkt (Quelle) bekannt. Dieser Ausgangspunkt besteht aus einem Rechnernamen (Internetadresse) und einer Portnummer. Portnummern ab 1024 können unter UNIX von jedem Programm frei belegt werden. Für die Verwendung einer Portnummer im Bereich von 1 bis 1023 benötigt ein Benutzer root-Privilegien. Mit einer Angabe “rootonly=true” akzeptiert ein Device Manager nur noch Verbindungen deren Ausgangspunkt-Portnummer unter 1024 liegt und beschränkt damit den Zugang auf Benutzer mit root-Privilegien. Der Standardwert von `rootonly` ist `false`.

dmuiaccess, dhaccess: legen fest, welche Arten von Verbindungen ein DM akzeptiert. "dmuiaccess=true" erlaubt Verbindungen von Benutzern zum DM, "dhaccess=true" dagegen erlaubt Verbindungen eines anderen DMs zur Datenübertragung. Standardwerte für dmuiaccess und dhaccess ist true.

chroot: Vor dem Start eines Systemkommandos (*Resource* vom Typ *SysCmd*) wird die Arbeitsumgebung (root- und Arbeitsverzeichnis) für dieses Kommando mit dem UNIX-Systemaufruf `chroot`¹³ gewechselt.

chdir: Beim Start eines Device Managers wechselt das Programms sein aktuelles Arbeitsverzeichnis in das angegebene Verzeichnis.

A.1.4.2 dev.conf

Die Datei `dev.conf` enthält eine Beschreibung jeder *Device-Resource*, die auf einem Rechner für den Device Manager verfügbar sein soll. Zusätzlich lassen sich Vorgabewerte (defaults) für *File*-, *Pipe*-, und *SysCmd-Resources* definieren.

PhysicalSizes: für jede *Resource* können Vorgabewerte für die Kommandos `setinput` und `setoutput` festgelegt werden. Die Bedeutungen von `block`, `buffer` und `modulo` wurden zusammen mit den `setinput`- und `setoutput`-Kommandos beschrieben.

format: wird zusammen mit dem Bandlabel verwendet, um ein eingelegtes Band zu überprüfen. Fehlt diese Angabe, dann findet der Name (*Identifier*) eines Eintrags Verwendung als Bestandteil eines Bandlabels. Formatangabe und Bandlabel ergeben zusammengesetzt einen Medienlabel.

dev: Pfadname, über den ein Bandlaufwerk angesprochen werden kann.

changer: Pfadname, über den ein Bandwechselroboter angesprochen werden kann.

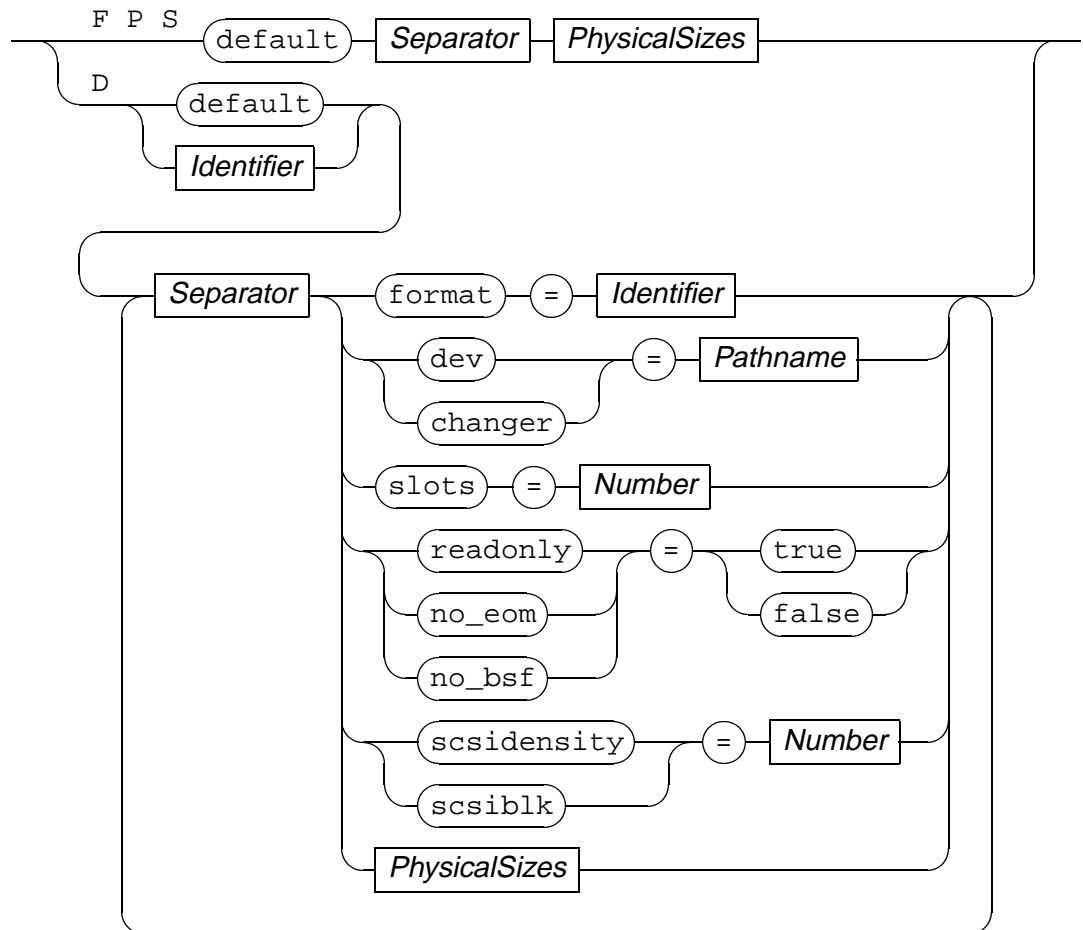
slots: Anzahl der Ablagefächer des Bandwechselroboters.

readonly: Mit "readonly=true" kann auf das betreffende Bandlaufwerk nur lesend zugegriffen werden¹⁴.

¹³Siehe UNIX-Manualseite zu `chroot(8)` und `chroot(2)`.

¹⁴Die Beschreibung von `readonly` in [DM] ist fehlerhaft.

dev.conf



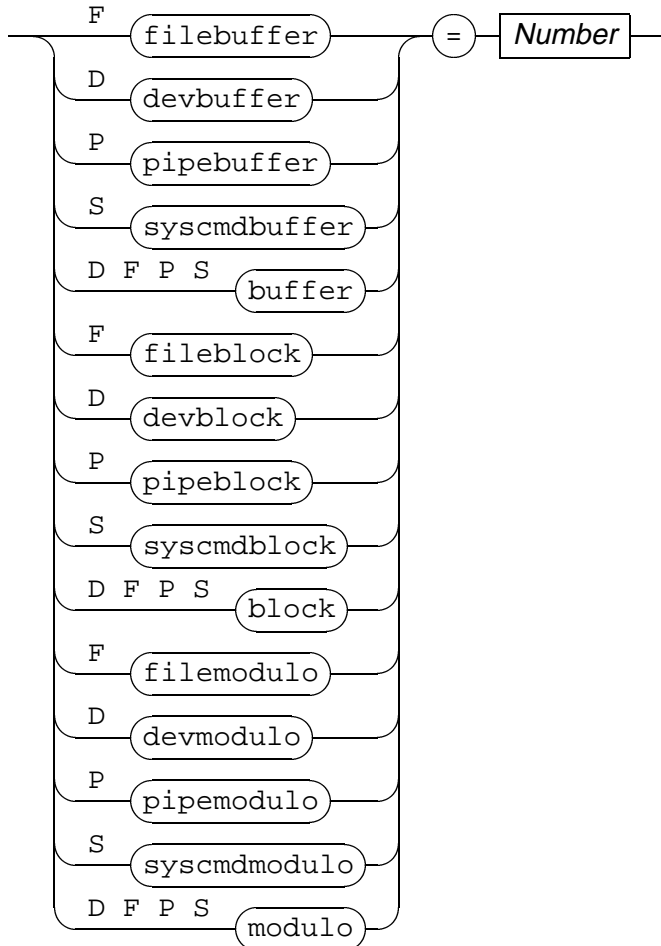
no_eom: Bei der Angabe von “no_eom=true” verwendet der Device Manager keine EOM-Marken (End Of recorded Media) auf Bandmedien¹⁵.

no_bsf: Zur Bandpositionierung verwendet der Device Manager u. a. BSF-Operationen¹⁶ (back space file). Einige Bandlaufwerke¹⁷ lassen diese Operation jedoch nicht zu. Mit der Angabe von “no_bsf=true” umgeht der DM die Einschränkung des Bandlaufwerks. Er spult das

¹⁵Einige Bandlaufwerke (z. B. 19-Zoll Spulenbandlaufwerke) kennen keine EOM-Marken. Sie verwenden stattdessen zwei unmittelbar aufeinanderfolgende EOF-Marken (End Of File).

¹⁶Siehe UNIX-Manualseiten zu `mt(7)`, `mt(8)` und `mtio(4)`, `mtio(7)`.

¹⁷Die meisten QIC-Laufwerke beherrschen keine BSF-Positionierung.

PhysicalSizes

Band an den Anfang zurück, um es dann von dort aus mit FSF-Operationen (forward space file) neu zu positionieren.

scsidensity: Fordert vom Bandlaufwerk die Umschaltung auf die angegebene Aufzeichnungsdichte. Aufzeichnungsdichten sind Zahlen im Bereich von 0 bis 255. Zulässige Angaben hängen vom verwendeten Laufwerkstyp ab und werden in der Dokumentation des Laufwerks aufgezählt¹⁸.

¹⁸Im SCSI-2-Standard sind einige Aufzeichnungsdichten als Werte definiert (<http://scitexdv.com/SCSI2/SCSI2-10.html#tab198>, Table 198 - Sequential-access density codes).

scsiblk: Fordert vom Betriebssystemtreiber für Bandlaufwerke die Verwendung einer bestimmten betriebssysteminternen Buffergröße zur Datenzwischenspeicherung an (device buffer)¹⁹.

Wird keine der Dateien `dev.conf.<hostname>` bzw. `dev.conf` beim Start des DM-Programms gefunden, stehen keine *Device-Ressourcen* zur Verfügung.

Das folgende Beispiel einer `dev.conf`-Datei enthält Angaben über zwei Bandlaufwerke. Jedes Bandlaufwerk verfügt über verschiedene Aufzeichnungsformate und Betriebsmodi. Diese sind im Beispiel jeweils mit einem separaten Eintrag aufgeführt.

```
# dev.conf
# Exabyte 8500
# reading/writing 8500 style tapes (5 GByte)
# ...with fixed record length of 1k
exa5f dev=/dev/scsi/rexabyte format=EXA5F
      scsidensity=21 scsiblk=1024 modulo=1024
# ...with variable record length
exa5v dev=/dev/scsi/rexabyte format=EXA5V
      scsidensity=21 scsiblk=0
# same tape drive
# reading/writing 8200 style tapes (2 GByte)
# ...with fixed record length of 1k
exa2f dev=/dev/scsi/rexabyte format=EXA2F
      scsidensity=20 scsiblk=1024 modulo=1024
# ...with variable record length
exa2v dev=/dev/scsi/rexabyte format=EXA2V
      scsidensity=20 scsiblk=0

# QIC Tape Streamer
# with restricted tape positioning capabilities
qic150 dev=/dev/scsi/rstreamer format=QIC150
       scsidensity=16 modulo=512 no_bsf=true
# QIC-150 Drives can read QIC-24 and QIC-11 format,
# but cannot write them!
qic24 dev=/dev/scsi/rstreamer format=QIC24
      scsidensity=5 modulo=512 no_bsf=true readonly=true
qic11 dev=/dev/scsi/rstreamer format=QIC11
      scsidensity=4 modulo=512 no_bsf=true readonly=true
```

¹⁹`no_eom`, `no_bsf`, `scsidensity` und `scsiblk` sind neue Funktionen des Device Managers, die in [DM] nicht beschrieben werden.

Zum Eintragen von Bandlaufwerken in der *smart*-Konfiguration stehen damit folgende *DM-Device-Ressourcen* zur Verfügung²⁰:

```
tapehost:Dexa5f  tapehost:Dexa2f  tapehost:Dqic150
tapehost:Dexa5v  tapehost:Dexa2v  tapehost:Dqic24
                                tapehost:Dqic11
```

Die in der Konfigurationsdatei aufgeführten Formatangaben werden zur Bildung von Medienlabel verwendet. Ein Bandmedium, das über ein *smart*-Benutzerinterface erstmalig angemeldet wurde, erhält als Label eine Aufschrift der Form “<format>-<laufende-Nummer>” zugewiesen, zum Beispiel “EXA5F-4711” oder “QIC150-4712”. So kann ein Operator anhand einer Beschriftung der Bandmedien leicht das verwendete Aufzeichnungsformat erkennen. Die von *smart* vergebenen Medienlabel sind eindeutig und dienen der Identifizierung einzelner Medien in Bandarchiv.

Die verwendeten Laufwerksnamen und Formatbezeichnungen sind frei wählbar. Jedoch ist eine Andeutung des Betriebsmodus eines Laufwerks über eine geeignete Wahl dieser Bezeichner empfehlenswert.

A.2 Benutzerinterface `xdmui`

`xdmui` ist ein X11-based Device Manager User Interface. Es bietet einen direkten Zugriff auf Funktionen des Device Managers. Über Formulare und Auswahlmenüs lassen sich DM-Kommandos schnell und komfortabel zusammenstellen.

Abbildung A.2 zeigt ein Fenster, wie es nach Start des `xdmui`-Programms auf eine X11-Bildschirm erscheint. Im oberen Teil sind Buttons und Menüs zu finden, über die sich verschiedene Kommandos zusammenstellen lassen. Darunter ist ein großes Textfeld angeordnet, das Ergebnisse der ausgeführten Kommandos anzeigt. Das in Abbildung A.2 dargestellte Message-Fenster enthält die Ausgabe eines `status`-Kommandos, welches vom Device Manager auf einem Rechner mit Namen `koppenas` ausgeführt wurde.

A.3 Benutzerinterface `dmcp`

`dmcp` ist ein Programm zur Verwendung auf einer Shell-Kommandozeile. Es dient zum einfachen Kopieren von Daten, die als *Resource*- oder *Alias*-Objekte vorliegen.

²⁰Der Rechner, an dem die Bandlaufwerke angeschlossen sind, heißt im Beispiel “tapehost”.

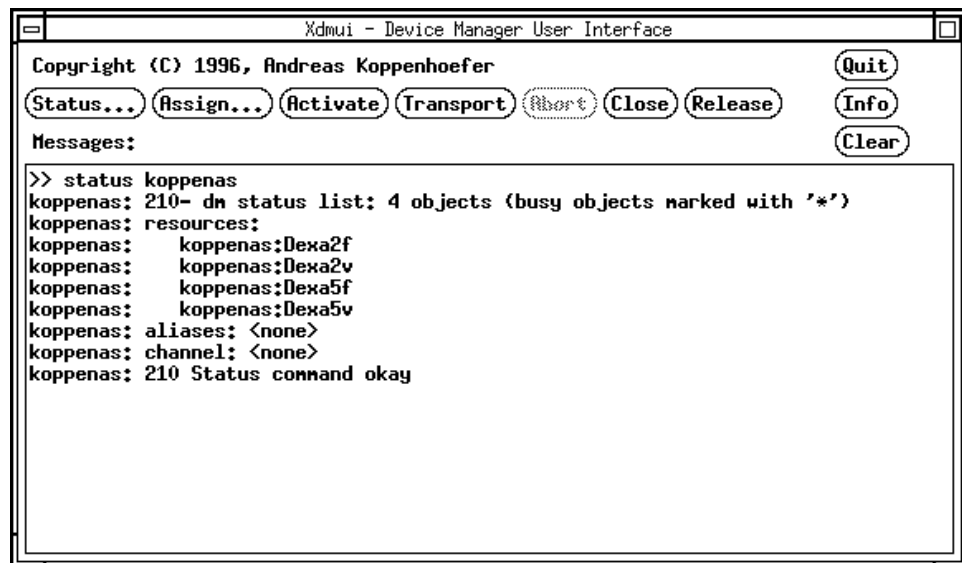


Abbildung A.2: xdmui: X11-based Device Manager User Interface, Hauptfenster

Anwendungsbeispiel: es soll ein Datencontainer von Band (*Resource* `koppenas:Dexa5f`, Bandlabel "H", file #9) in ein Verzeichnis einer Festplatte kopiert werden (*Resource* `koppenas:F/tmp/archive.tar`). Mit den Benutzerinterfaces `dmui` und `xdmui` sind dazu mehrere Arbeitsschritte nötig. Zu den Kommandos werden jeweils alle Ausgaben²¹ des Device Managers gezeigt, da diese zum Teil (im Beispiel unterstrichen) für die nachfolgenden DM-Kommandos benötigt werden. Kommandozeilen sind zur besseren Lesbarkeit mit dem Zeichen ">" markiert.

```
> assign koppenas:Dexa5f label=H
  111-Resource:                koppenas:Dexa5f
  111 Assigned alias:          koppenas:AH-5433-9
  115 Checking tape label:     koppenas:Dexa5f
  201 Alias ready:             koppenas:AH-5433-9
> assign koppenas:F/tmp/archive.tar create
  111-Resource:                koppenas:F/tmp/archive.tar
  111 Assigned alias:          koppenas:Afile-5435-7
  201 Alias ready:             koppenas:Afile-5435-7
> setinput koppenas:AH-5433-9 file=9
  112-Alias:                   koppenas:AH-5433-9
  112 Assigned channel:        koppenas:Cin1335-5436-6
  115 positioning tape:        koppenas:Dexa5f
```

²¹DM-Antworten wurden ähnlich dem FTP-Protokoll gestaltet ([RFC 959]). Ihre Bedeutungen beschreibt [DM, Kapitel 3.4].

```

155 label EXA5F-H, file 9: koppenas:Cin1335-5436-6
202 Channel ready:          koppenas:Cin1335-5436-6
> setoutput koppenas:Afile-5435-7
112-Alias:                  koppenas:Afile-5435-7
112 Assigned channel:       koppenas:Cout1336-5437-4
202 Channel ready:          koppenas:Cout1336-5437-4
> transport koppenas:Cin1335-5436-6 koppenas:Cout1336-5437-4
120-Pending command transport: koppenas:Cin1335-5436-6
120 wait for final message
121 Starting transport:      koppenas:Cin1335-5436-6
121 Starting transport:      koppenas:Cout1336-5437-4
142 196608 bytes read:       koppenas:Cin1335-5436-6
214 Finished & closed:       koppenas:Cin1335-5436-6
142 196608 bytes written:    koppenas:Cout1336-5437-4
213 Transport finished:      koppenas:Cout1336-5437-4
> close koppenas:Cout1336-5437-4
120-Pending command close:   koppenas:Cout1336-5437-4
120 wait for final message
212 Channel closed:          koppenas:Cout1336-5437-4
> release koppenas:AH-5433-9 eject
120-Pending command release: koppenas:AH-5433-9
120 wait for final message
215 Alias released:          koppenas:AH-5433-9
> release koppenas:Afile-5435-7
200-Object:                  koppenas:Afile-5435-7
200 Command okay

```

Es ist deutlich zu erkennen, daß die direkte Benutzung der DM-Kommandoschnittstelle sehr aufwendig sein kann: für dieses einfache Beispiel sind bereits acht Kommandos notwendig. `dmcp` vereinfacht das Beispiel auf eine einzige Shell-Kommandozeile:

```
dmcp koppenas:Dexa5f label=H file=9 eject -- \
      koppenas:F/tmp/archive.tar create
```

Zu Argumenten für `dmcp` faßt man alle Objekte und Kommandooptionen eines Transports nach Quelle und Ziel getrennt zusammen (aus dem Beispiel: alle nicht unterstrichenen Kommandoargumente). Diese zusammengefaßten Objekt- und Optionenlisten werden im `dmcp`-Kommando aufgezählt. Ein doppelter Trennstrich "--" steht dabei zwischen Quellenangaben auf der linken und Zielangaben auf der rechten Seite.

Ohne Argumente gibt `dmcp` einen kurzen Hilfetext aus:

```
Usage: dmcp [ -v ] [ -d ] [ -p default-port ] \  
  input-resource-or-alias [input-options] [source-port] [...] -- \  
  output-resource-or-alias [output-options] [target-port] [...]  
Options: -v  verbose: display all replies from dm.  
         -d  debugging: prints a lot of unreadable messages.
```

Quelle und Ziel müssen als *Resource*- oder *Alias*-Objekte angegeben werden. Zu jedem angegebenen Objekt dürfen Optionen hinzugefügt werden. `dmcp` akzeptiert die Angabe mehrerer Quellen und Ziele. Die Daten der Quellobjekte werden alle nacheinander auf das erste Zielobjekt kopiert. Für den Fall, daß diese Kopieroperation abbricht, schaltet `dmcp` die Ausgabe auf das nächste Zielobjekt um, sofern ein solches vorhanden ist. Dieses Umschalten auf alternative Ziele wird solange wiederholt, bis alle Daten kopiert werden konnten oder keine Ziele mehr zur Verfügung stehen.

Abschnitt 5.2.3 enthält ein weiteres Beispiel für den Einsatz von `dmcp`.

Anhang B

Formular “Anforderung einer Restaurierung”

Zur Restaurierung von Daten benötigt der Datensicherungsoperator von Ihnen verschiedene Angaben. Ohne diese Hilfe ist es unmöglich aus der Menge der gesicherten Daten gezielt die richtigen herauszufinden. Wenn Sie einzelne Fragen nicht genau beantworten können, sind ungefähre Angaben meist sehr hilfreich. Die Angaben in Klammern dienen als Hilfestellung zum Ausfüllen des Fragebogens.

1. Was soll restauriert werden: einzelne Dateien, Verzeichnisse, Partition (Filesystem) oder eine komplette Festplatte?
2. Von welcher Art sind die Daten (Benutzerdaten, Datenbank, Betriebssystem, etc.)?
3. Wurden die Daten zerstört, verfälscht oder gelöscht und aus welchem Grund?
4. In welchem Verzeichnis befanden sich die Dateien bzw. Verzeichnisse (Ausgabe des Kommandos `'pwd'` in diesem Verzeichnis)? Bitte vollständige Pfade angeben.
5. Auf welchem Rechner befanden sich die Daten (Ausgabe der Kommandos `'uname -a'` oder `'hostname'`)?
6. Auf welchem Filesystem befanden sich die Daten (Ausgabe der Kommandos `'df .'`, `'df -k .'` oder

- 'bdf .' in diesem Verzeichnis; Kommando ist vom Betriebssystem abhängig)?
7. Wann wurden die Daten zuletzt verändert (Der Datensicherungsoperator wird versuchen den Zustand der Daten nach diesem Datum herzustellen.)?
 8. Wann wurden die Daten erstmalig erzeugt?
 9. Wann wurden die Daten zerstört/verfälscht/gelöscht?
 10. Wieviel Speicherplatz nehmen die gewünschten Daten in Anspruch?
 11. Wohin sollen die Daten restauriert werden? (Für kleine Datenmengen: restaurierte Benutzerdaten werden vom Operator sinnvollerweise im Verzeichnis ~/restore abgelegt. Dazu müssen Sie als Benutzer die Kommandos 'mkdir ~/restore', 'chmod 777 ~/restore' und 'chmod a+x ~' ausführen, damit der Datensicherungsoperator die Daten in diesem Verzeichnis ablegen kann.)
 12. Falls die Daten an den ursprünglichen Platz restauriert werden sollen: Befinden sich dort bereits Daten und, wenn Ja, dürfen diese überschrieben, ersetzt werden? Der Datensicherungsoperator benötigt dazu Schreibrechte in den entsprechenden Verzeichnissen.
 13. Wie wichtig sind die Daten? Ist die Restaurierung dringend?
 14. Wie aufwendig ist die Arbeit, die Daten neu zu erzeugen, falls der Datensicherungsoperator nicht in der Lage ist, diese zu restaurieren.
 15. Sind Sie telefonisch oder per Email für eventuelle Rückfragen erreichbar? Wenn Ja, wann und wie (Arbeitszeit, Telefonnummer, Email-Adresse)?
 16. Verfügen Sie über root-Privilegien? Wenn nein, wer ist der zuständige Administrator des Rechners, auf dem die Daten restauriert werden sollen?
-

Anhang C

Statistiken

Statistiken dieser Diplomarbeit beruhen auf Daten, die von Rechnern der Fakultät Informatik der Universität Stuttgart und auf dem privaten Rechner des Autors gesammelt wurden¹.

C.1 Änderungshäufigkeit von Daten

Statistiken, die die Änderungshäufigkeit von Daten betreffen wurden im Zeitraum Dezember 1995 bis November 1996 auf folgenden Rechnern ermittelt.

Host	Typ, Betriebssystem	Benutzer	Hauptnutzung
trick	sun10/42, Solaris 2.x	1300	Server für Rechnerpool Grundstudium
zdi	sun20/712, Solaris 2.x	20	Server für Usenet; Ingres-Datenbank (Bibliothekskatalog)
inf	sun10/40, Solaris 2.x	10	Server für FTP, WWW und Backup
ako	i486DX2-66 PC, Linux	1	privater Rechner des Autors; Entwicklung von <i>smart</i>

Bei der Auswahl der Rechner wurde darauf geachtet, möglichst verschiedene Nutzungsarten abzudecken. Somit konnten aussagekräftige statistische Daten zusammengetragen werden.

Für die in Abbildung 3.1 dargestellten Datentypen wurden folgende Filesystem ausgewählt.

¹Die gesammelten statistischen Daten werden vom Autor auf Anfrage zur Verfügung gestellt.

Datentyp	Filesystem
Mail	trick:/var/mail
News	zdi:/news/spool/news
Home	ako:/home
System	zdi:/usr

Abbildung 3.2 enthält das Ergebnis einer Simulation von Datensicherungen mit unterschiedlichen Sicherungssequenzen und Schwellwerten. Einzelne Filesysteme wurden entsprechend ihrem Anteil am Umfang aller Sicherungen gewichtet, d. h. statistische Angaben großer Filesysteme haben mehr Einfluß auf das Ergebnis als die kleinerer Filesysteme.

Für Abbildung 3.3 wurden statistische Angaben aller Daten (Filesysteme) nach Datentyp getrennt bewertet. Diese Werte wurden entsprechend ihrem Anteil an allen Sicherungen gewichtet.

C.2 Betriebsstatistik

Seit Anfang 1990 setzt die Fakultät Informatik zentrale Datensicherung ein. Zu diesem Zweck wurde die inkrementelle Sicherungsmethode 9 mit einer Sicherungssequenz von 05555 35555 ausgewählt².

Als Datensicherungsmedium kommt ein Laufwerk vom Typ Exabyte 8500 mit 5 GByte Kapazität zum Einsatz, welches über einen Bandwechselroboter EXB-10e (Magazin für 10 Medien) bedient wird.

Anfang November 1996 waren 86 Hosts mit insgesamt 465 Filesystemen zur Datensicherung angemeldet. Die verfügbare Plattenkapazität betrug 240 GByte, wovon 149 GByte (62%) mit Daten belegt waren.

In Kapitel 1 aufgeführte Betriebsstatistiken enthalten Daten, die während des Betriebs der zentralen Datensicherung gesammelt wurden.³

In Abbildung 3.4 ist für jeden Client und Filesystem die maximal erreichte Übertragungsgeschwindigkeit eingetragen. Ausgewertet wurden Datensicherungen des Zeitraums vom 6. Oktober bis 4. November 1996.

²Sicherungssequenz: siehe Sicherungsmethode 9

³Die statistischen Daten werden vom Autor oder vom Administrator der zentralen Datensicherung auf Anfrage zur Verfügung gestellt.

Anhang D

Bezugsquellen

Zum Einsatz von *smart* werden verschiedene frei erhältliche Softwarepakete benötigt. Die Bezugsquellen¹ sind als URLs (siehe Glossar, Seite 117) angegeben. Informationen über Konfiguration und Installation enthalten den jeweiligen Pakete.

Software zum Betrieb des Device Managers aus [DM] für Clients ist im *smart*-Softwarepaket enthalten. Auf dem zentralen Backup-Server benötigt der Volume Manager folgende Softwarepakete.

D.1 Datenbank

Der Volume Manager benutzt eine SQL-Datenbank. Mini-SQL (mSQL) beherrscht als Datenbank nur einen kleinen Teil der SQL-Anfragesprache (ANSI-Standard). Diese Software darf zum nicht-kommerziellen Einsatz frei verwendet werden. Kommerziell ausgerichtete Organisationen und Unternehmen dürfen mSQL gegen eine geringe Lizenzgebühr ebenfalls nutzen. Bezugsquellen der Software und weitergehende Informationen über mSQL:

Software

`ftp://bond.edu.au/pub/Minerva/msql/msql-1.0.16.tar.gz`

Informationen (FAQ)

`ftp://bond.edu.au/pub/Minerva/msql/faq.html`

`http://www.swl.fh-heilbronn.de/msql`

Das Perl-Modul `MsqlPerl` (siehe unten) enthält in der Datei `"patch.lost.tables"` eine wichtige Fehlerkorrektur für die mSQL-Version 1.0.16. Ein entsprechender Hinweis ist auch in der FAQ zu finden.

¹Stand November 1996

D.2 Perl und Perl-Module

Perl und die meisten Perl-Erweiterungsmodule sind über viele sogenannte CPAN-Mirror² erhältlich. Dazu gehören in Deutschland unter anderem...

```
ftp://ftp.leo.org/pub/comp/programming/languages/perl/CPAN/
ftp://ftp.rz.ruhr-uni-bochum.de/pub/CPAN/
ftp://ftp.uni-hamburg.de/pub/soft/lang/perl/CPAN/
```

Die folgenden Verzeichnisangaben beschreiben relative Pfade unterhalb eines CPAN-Mirrors.

Paket/Modul	CPAN-Mirror-Pfad File	benötigt von
Perl-5	authors/id/ANDYD/ perl5.003_07.tar.gz	VM, dmcp, xdmui
CGI.pm	modules/by-module/CGI/ CGI.pm-2.27.tar.gz	VM
MsqlPerl	modules/by-module/Msql/ MsqlPerl-1.11.tar.gz	VM
Term:: ::ReadLine	modules/by-module/Term/ Term-ReadLine-0.92.tar.gz	pmsql
::ReadKey	TermReadKey-2.05.tar.gz	
Sx	modules/by-module/Sx/ Sx-2.2.tar.gz	xdmui

Die aufgezählten Module werden nur auf dem Rechner vorausgesetzt, auf dem die *smart*-Software des Volume Managers (VM) eingesetzt werden soll. Für das Benutzerinterface *dmcp* genügt eine ältere Perl-Version (perl-4.03x). *pmsql* ist eine interaktive Shell für den direkten Zugang zur *mSQL*-Datenbank (im Datenbankmodul *MsqlPerl* enthalten).

Im allgemeinen werden Perl-Module mit folgender Kommandosequenz installiert:

```
gzip -dc <PerlModule>.tar.gz | tar xpf -
cd <PerlModule>
perl Makefile.PL
make
make test
make install
```

Weitergehende Installationshinweise enthalten Dateien mit Namen `README` und `INSTALL` aus den jeweiligen Modulen.

²Comprehensive Perl Archive Network (CPAN) – Verschiedene FTP-Server mit Software und Information zur Programmiersprache Perl

D.3 WWW-Server

Die Benutzerschnittstellen des Volume Managers wurden als HTML-Dokumente und CGI-Skripte gestaltet. CGI-Skripte startet ein WWW-Server auf Anforderung eines Benutzers. Sie wurden entwickelt und getestet mit dem Apache-WWW-Server Version 1.1.1, der über die URL <http://www.apache.org/> erhältlich ist. Für eine Zugangskontrolle (Autorisierung) zur Benutzerschnittstelle kommt das Apache-Modul `mysql_auth_module` zum Einsatz.

Die Verwendung eines Apache-WWW-Server für den Betrieb von *smart* ist jedoch nicht unbedingt erforderlich. Die Benutzerschnittstelle lässt sich leicht für einen anderen WWW-Server und dessen Zugangskontrolle anpassen. Informationen darüber sind in den Quelltexten von *smart* im CGI-Skript `user-admin.pl` enthalten.

Weitergehende Informationen zu Installation und Konfiguration des WWW-Servers und der CGI-Skripten sind der Dokumentation des WWW-Servers und dem *smart*-Softwarepaket zu entnehmen.

Glossar

ACL: Access-Control-List (ACL), Liste von Gruppen zur Festlegung von Zugriffsrechten.

Administrator: Benutzer, der Dienste der zentralen Datensicherung nutzen kann. Zu diesen Diensten gehören u. a. Abruf von Informationen über die Datensicherung, Anforderung einer Restaurierung. Administratoren können Gruppen zugeordnet werden. Der Zugriff eines Administrators ist beschränkt auf Daten und Informationen der zu diesen Gruppen gehörenden Clients.

Alias: Objekttyp des Device Managers. Entsteht bei einer Anmeldung eines *Resource*-Objekts.

Bandwechselroboter: Roboter, der Medien aus einem Magazin in ein Bandlaufwerk einlegen und wieder entnehmen kann. Das Magazin hat in der Regel Lagerplätze für eine kleine Anzahl Medien. Die Einheit von Magazin und Roboter bezeichnet man auch als Jukebox.

Benutzerinterface: Programme zur Verwendung durch Administratoren.

CGI: Common Gateway Interface (CGI), Schnittstelle für dynamische Hypertextdokumente, die auf Anfrage von CGI-Skripten generiert werden. Wird von WWW-Servern verwendet.

Channel: Objekttyp des Device Managers. Ein *Alias*-Objekt läßt sich für einen Datentransport vorbereiten und als Kanal öffnen.

Client: Host, der zur regelmäßigen Datensicherung angemeldet ist. In der Regel ist ein Server zugleich auch Client.

Console: Operatorinterface, über das die Software der zentralen Datensicherung Informationen und Fehlermeldungen ausgibt.

daemon: Ständig laufendes oder bei Bedarf automatisch startendes Programm; bietet verschiedene Dienste an. WWW oder FTP werden z. B. über daemon abgewickelt.

Datencontainer: siehe Volume.

DM: Device Manager (DM), Programm (daemon), das Daten im Netzwerk transportieren und auf Bandlaufwerke (devices) zugreifen kann.

EOF: End Of File (EOF) kennzeichnet das Ende einer Datei. Da Bandlaufwerke nur einen sequentiellen Zugriff auf die Daten eines Mediums erlauben, dienen sogenannte EOF-Marken zur Trennung einzelner Aufzeichnungen. Diese Aufzeichnungen nennt Files. Sie werden bei Null beginnend vom Bandanfang aus fortlaufend nummeriert.

EOM: End Of Media (EOM), bezeichnet meist das Ende von auf Band aufgezeichneten Daten. Viele Bandlaufwerke schreiben nach der letzten Aufzeichnung eine sogenannte EOM-Marke auf das Band. EOM bedeutet in diesem Fall "End Of *recorded* Media". EOM wird auch als Synonym für "End Of Tape" (EOT) verwendet, welches das Ende des (für Aufzeichnungen nutzbaren) Bandmediums bezeichnet.

FIFO: Eine in [LEW 91] verwendete Bezeichnung für UNIX-Pipes.

Filesystem: Häufig verwendetes Synonym für zu sichernde Daten eines Client. Gemeint ist damit ein Teil des Datenbestands, der von den unter UNIX üblichen Datensicherungsprogramme (z. B. `dump`) gemeinsam behandelt wird. Diese müssen aber nicht notwendigerweise als Filesystem (Partition einer Festplatte) organisiert sein.

Fragment: Teil eines Volumes (Datencontainer); entsteht bei der Aufteilung eines Volumes in kleinere Stücke.

FTP: File Transfer Protocol (FTP), Protokoll zur Übertragung von Dateien im Internet.

GByte: Maßeinheit für Umfang bzw. Größe von Daten;

$$1 \text{ GByte} = 1024 \text{ MByte} = 1048576 \text{ KByte} = 1073741824 \text{ Byte} = 2^{30} \text{ Byte}$$

Gruppe: Clients und Benutzer der zentralen Datensicherung können unter *smart* Gruppen zugeordnet werden. Über diese Gruppen lassen sich u. a. Zugriffsrechte bestimmen.

Host: Rechner, Computer mit Anschluß an ein Netzwerk.

HP-UX: System V UNIX Version der Firma Hewlett Packard.

HTML: Hypertext Markup Language (HTML) ist die Sprache in der Hypertextdokumente geschrieben werden. Es ist eine Untermenge von SGML und beinhaltet Mechanismen zur Verknüpfung verschiedener Dokumente.

http: Hypertext Transfer Protocol (http), Protokoll zur Übertragung von Hypertextdokumenten im Internet.

Hyperlink: Ein Verweis innerhalb eines Hypertextdokuments auf ein anderes Dokument, das selbst ein Hypertextdokument sein darf.

Hypertext: Ein Dokument, das in HTML geschrieben wurde und Hyperlinks zu anderen Dokumenten enthält. Hypertextdokumente werden üblicherweise über das WWW verteilt.

Intranet: Bezeichnung für ein internes, lokales Netzwerk eines großen Unternehmens oder Institution.

Jukebox: siehe Bandwechselroboter.

KByte: Maßeinheit für Umfang bzw. Größe von Daten;

$$1 \text{ KByte} = 1024 \text{ Byte} = 2^{10} \text{ Byte}.$$

Label: dient der Unterscheidung von Bandaufzeichnungen. *smart* verwendet drei verschiedene Labeltypen:

Bandlabel: kurzer Datenblock, den ein Device Manager an den Anfang eines neuen Bands schreibt. Enthält u. a. Zeitangaben, Bezeichnung des Bandformats, Medienlabel.

Filelabel: kurzer Datenblock, der vor jedem Datencontainer auf Band geschrieben wird. Sein Inhalt ist dem des Bandlabels ähnlich. Enthält u. a. Informationen über den Datencontainer.

Medienlabel: Identifikationsmerkmal eines Mediums; setzt sich aus der Bezeichnung des Bandformats und einer Mediennummer zusammen. Wird unter *smart* zum An- und Abmelden von Medien verwendet.

Linux: Frei erhältliches UNIX-Betriebssystem, ursprünglich nur auf Intel Prozessoren basierenden Rechnern (PC), inzwischen auch auf DEC Alpha und SPARC Plattformen verfügbar.

MByte: Maßeinheit für Umfang bzw. Größe von Daten;

$$1 \text{ MByte} = 1024 \text{ KByte} = 1\,048\,576 \text{ Byte} = 2^{20} \text{ Byte}.$$

News: siehe Usenet.

Operator: Administratoren der Gruppe "Operator" haben Zugriff auf alle gesicherten Daten und Inhaltsverzeichnisse, sowie deren Konfiguration. Operatoren sind auch für die Bestückung der Bandlaufwerke mit Medien zuständig und haben Zugang zum Bandarchiv.

Operatorinterface: Programme zur Verwendung durch Administratoren der Gruppe "Operator".

Pipe: spezieller Dateityp unter UNIX. Daten, die von einem Prozeß in eine Pipe (Kanal) geschrieben werden, kann ein anderer Prozeß wieder in der gleichen Reihenfolge (FIFO, "first-in-first-out") auslesen. Eine UNIX-Pipe, die über einen Pfadnamen zugänglich ist, wird auch als "named pipe" bezeichnet.

Prozeß: laufendes, aktiviertes Programm.

Resource: Menge von Objekten eines Device Managers. *Resource*-Objekte können bei dem für das Objekt zuständigen Device Manager angemeldet werden und stehen danach für weitere Operationen als *Alias*-Objekt zur Verfügung. Der englische Begriff *Resource* und alle anderen DM-Objektbezeichner (*Alias*, *Channel*) werden in dieser Dokumentation *kursiv* wiedergegeben.

Ressource: Der deutsche Begriff "Ressource" steht als Synonym für zur Verfügung stehende Arbeitsmittel bzw. Kapazitäten der zentralen Datensicherung.

RFC: Request for Comments (RFC), online über FTP, Kermit oder Electronic Mail von `NIC.DDN.MIL` und anderen Fileservern³ abrufbare Texte. Enthält Beschreibungen von Standards und Protokollen des Internet.

Server: Zentraler Rechner bzw. Host, auf dem die Software der zentralen Datensicherung läuft.

SGML: Standardized Generalized Markup Language (SGML) ist ein internationaler Standard zur Definition von system- und hardwareunabhängigen Methoden zur Darstellung von Text in elektronischen Medien.

smart: Abkürzung bestehend aus den Anfangsbuchstaben von "System-Manager für Archivierung, Restaurierung und Transport", Konzepte und Software zur Durchführung von regelmäßigen, automatischen Datensicherungen einer großen Anzahl von Maschinen eines Intranet.

Solaris: Name eines Produkts der Firma SunSoft, das unter anderem das Betriebssystem SunOS enthält.

SunOS: UNIX Version der Firma Sun/SunSoft. Seit der Version SunOS 5.0 ist auch die Bezeichnung Solaris gebräuchlich.

SYSV: System V, alte UNIX-Version der Bell Laboratories. Wird heute als Kennzeichen einer Familie von UNIX-Versionen mit gemeinsamen Eigenschaften verwendet.

³<ftp://ftp.rus.uni-stuttgart.de/pub/doc/standards/rfc/>

UNIX: Betriebssystemname; die ersten UNIX-Versionen stammen von Ken Thompson und Dennis Ritchie der AT&T Bell Laboratories.

URL: Uniform Resource Locator (URL), eine kompakte Beschreibung von Ressourcen, die über Internet abgerufen werden können, als Zeichenkette dargestellt. URLs werden in erster Linie zum Abrufen von Dokumenten und Informationen im WWW verwendet. Syntax und Semantik ist in [RFC 1738] definiert. Meistens enthält eine URL Informationen über das Zugriffsprotokoll, die Adresse des Servers und weitere Details, wie z. B. einen Pfad oder Filenamen.

Usenet: Eine Sammlung von mehreren tausend Diskussionsgruppen, deren Artikel (News) über Internet und andere Netzwerke verteilt werden. Usenet-Teilnehmer können eigene Artikel veröffentlichen und Artikel anderer Teilnehmer lesen. Zum Usenet gehören sogenannte News-Server, die die Artikel verteilen und zum Abruf in einem News-Spool bereithalten.

VM: Volume Manager (VM), Programm (daemon), das selbständig und unbeaufsichtigt die Datensicherung einer Vielzahl von Clients steuert und über Inhaltsverzeichnisse der gesicherten Daten deren Restaurierung ermöglicht.

Volume: Auf Clients erzeugen Datensicherungsprogramme einen Strom von Bytes, der die gesicherten Daten enthält. Dieser Datenstrom mit einem definiertem Anfang und Ende wird als Datencontainer oder Volume bezeichnet. Jedes Volume erhält eine eindeutige Nummer (Identifier, ID). Anhand dieser sogenannten Volumenummer oder Volume-ID lassen sich einzelne Datencontainer unterscheiden.

WWW: World Wide Web, ein auf Hypertext basierendes verteiltes Informationssystem.

Weitere Begriffe und Abkürzungen finden Sie in [RFC 1983].

Literaturverzeichnis

- [AHO 88] Aho, Alfred V.; Sethi, Ravi; Ullman, Jeffrey D. : *Compilerbau, Teil I*. Bonn u. a. : Addison-Wesley, 1988.
- [BA 94] Bates, Regis J. : *Disaster recovery for LANs; a planning and action guide*. McGraw-Hill, 1994.
- [CGI] Gundavaram, Shishir : *CGI Programming on the World Wide Web* (Nutshell Handbook). Sebastopol, Calif. : O'Reilly, 1996.
- [DM] Koppenhöfer, Andreas : *Implementierung eines Datentransportservices in einem heterogenen Computer-Netzwerk* (Studienarbeit Nr. 1298). Universität Stuttgart, Institut für Informatik, 24. März 1994, online verfügbar.⁴
- [HSM] Bachmann, Jörg : *Automatisch sicher; HSM – Hierarchical Storage Management*. IN: iX Multiuser-Multitasking-Magazin, Oktober 1996, S. 106 ff.
- [HTML] Musciano, Chuk; Kennedy, Bill : *HTML – The definitive Guide* (Nutshell Handbook). Sebastopol, Calif. : O'Reilly, 1996.
- [IT 96] *IT-Grundschutzhandbuch 1996 – Maßnahmenempfehlungen für den mittleren Schutzbedarf* (Schriftenreihe zur IT-Sicherheit ; 3). Bundesamt für Sicherheit in der Informationstechnik, Bundesanzeiger : Köln, 1996.
- [LA 96] Langer, Volker : *Kleiner Bandsalat; Backup-Techniken im Vergleich: Leistungspotential und Sicherheit*. IN: iX Multiuser-Multitasking-Magazin, Oktober 1996, S. 124 ff.
- [LEW 91] Lewine, Donald A. : *POSIX Programmer's Guide : Writing Portable UNIX Programs with the POSIX.1 Standard*. Sebastopol, Calif. : O'Reilly, 1991.

⁴<http://www.informatik.uni-stuttgart.de/menschen/ako/dm.html>

- [Perl-5] Wall, Larry; Christiansen, Tom; Schwartz, Randal L. : *Programming Perl* (Nutshell Handbook). Sebastopol, Calif. : O'Reilly, 2nd Edition, 1996.
- [POSIX.1] Institute of Electrical and Electronics Engineers (IEEE) : *Standard for Information Technology – Portable Operating System Interface (POSIX). Part 1: System Application Programming Interface (API)*. 1990.
- [POSIX.2] Institute of Electrical and Electronics Engineers (IEEE) : *Standard for Information Technology – Portable Operating System Interface (POSIX). Part 2: Shell and Utilities*. 1992.
- [RFC 854] Postel, J.; Reynolds, J. : *Telnet Protocol Specification* (Request for Comments ; RFC 854). University of Southern California, Information Sciences Institute : May 1983, online verfügbar.⁵
- [RFC 959] Postel, J.; Reynolds, J. : *File Transfer Protocol* (Request for Comments ; RFC 959). University of Southern California, Information Sciences Institute : October 1985, online verfügbar.⁶
- [RFC 1738] Masinter, L.; Berners-Lee, T.; McCahill, M. : *Uniform Resource Locators (URL)* (Request for Comments ; RFC 1983). Network Working Group; University of Minnesota; December 1994, online verfügbar.⁷
- [RFC 1983] Malkin, G. : *Internet Users' Glossary* (Request for Comments ; RFC 1983). August 1996, online verfügbar.⁸
- [SQL 93] Date, C. J.; Darwen, Hugh : *A Guide to Sql Standard*. Addison-Wesley, 3rd Edition, 1993.
- [ST 95] Stevens, W. Richard : *Programmierung in der UNIX-Umgebung: die Referenz für Fortgeschrittene*. Bonn, Paris : Addison-Wesley, 1995.

⁵<ftp://ftp.rus.uni-stuttgart.de/pub/doc/standards/rfc/RFC800/rfc854.txt.gz>

⁶<ftp://ftp.rus.uni-stuttgart.de/pub/doc/standards/rfc/RFC900/rfc959.txt.gz>

⁷<ftp://ftp.rus.uni-stuttgart.de/pub/doc/standards/rfc/RFC1700/rfc1738.txt.gz>

⁸<ftp://ftp.rus.uni-stuttgart.de/pub/doc/standards/rfc/RFC1900/rfc1983.txt.gz>

Index

- abort, 91
- Abteilung, 16, 18, 25, 39, 45, 69, 105
- Access-Control-List, *siehe* ACL
- ACL, 39, 71, 73, 74, 113
- add, 92
- Administrator, 16–18, 27, 57, 69, 105, 113
- Alias*, 27, 79–103, 113
- all, 93
- Anwendungsgebiet, 78
- Archivierung, 9, 57
 - Archivierungsfrist, 39, 74
 - Aufgabenbereich, 25
- assign, 85
- AssignOptions*, 87
- Attributes*, 88
- Aufbewahrungsfrist, *siehe* Archivierungsfrist
- Aufgabenbereiche, 24
- Automat, 65
- Autorisierung, 67

- Backup
 - Aufgabenbereich, 24
- Bandlabel, *siehe* Label
- Bandlaufwerk, 18, 20, 24, 28, 73
 - Dimensionierung, 21
- Bandwechselroboter, 73, 83, 86, 92, 96, 113
- Benutzer, 17, 27, 79–96
 - interface, 17, 24, 26–27, 44, 77, 79, 100–103, 110, 113
 - schnittstelle, *siehe* Benutzerinterface
 - verwaltung, 69
- Besitzer, 88, 92
- block, 89
- Browser, 44
- buffer, 89

- Cache, 28–30, 71
- CGI, 110, 111, 113, 119
- changer, 96
- Channel*, 80–102
- chdir, 96
- chroot, 96
- Client, 17, 23, 25, 27, 109, 113
- close, 91
- Command*, 85, 86
- Common Gateway Interface, *siehe* CGI
- configuration*, 93
- Console, 18, 24, 43, 113
- create, 87

- daemon, 113, 117
- Datenbank, 13, 23, 24, 44, 46, 51, 63, 66, 68, 78, 109
- Datencontainer, *siehe* Volume
- Datensicherung, *siehe* Sicherung
- Datentyp, 5–8, 14
- del, 92
- delete, 91
- dev, 96
- dev.conf, 83, 96, 97
- Device Manager, *siehe* DM
- Devices
 - Aufgabenbereich, 24
- dhaccess, 95
- Digit*, 81

- DM, 25–27, 59, 77, 79–103, 109, 114, 119
 - Benutzerschnittstelle, *siehe* Benutzerinterface
 - Kommandos, 85, 100, 101
 - Konfigurationsfiles, 93
 - Objektnamen, 83
- dm.conf, 93–95
- dmcp, 26, 52, 100, 110
- xdmui, 110
- dmui, 26
- dmuiaccess, 95
- dump
 - level, 10, 41, 58
 - sequence, *siehe* Sequenz
- dump, 10, 15, 40, 56
- Dump-ID, 42
- eject, 92
- End Of File, *siehe* EOF
- EOF, 72, 97, 114
- EOM, 97, 114
- FIFO, 83, 114, 116
- file, 89
- File Transfer Protocol, *siehe* FTP
- Filelabel, *siehe* Label
- Filesystem, 17, 73, 78, 114
- format, 96
- Fragebogen, 105
- Fragment, 17, 28–31, 49, 52, 89, 114
- FTP, 101, 110, 114, 120
- GByte, 114
- group, 88
- Gruppe, 17, 39, 88, 113, 114
- Home, *siehe* Datentyp
- Host, 73, 114
- Hostname, 82
- HP-UX, 114
- HTML, 114, 119
- http, 115
- Hyperlink, 115
- Hypertext, 115
 - Markup Language, *siehe* HTML
 - Transfer Protocol, *siehe* http
- Identifier, 81
- info, 88, 89
- input, 87
- Internet Glossary, 120
- Intervall, 15, 20, 21, 31, 33–36, 73
- Intranet, 16, 115
- Jukebox, *siehe* Bandwechselroboter
- KByte, 115
- KeyValuePair, 94
- Kommandoschnittstelle, *siehe* Benutzerinterface
- Konfiguration, 27
 - Aufgabenbereich, 25
- Konfigurationsfiles, 93
- Kontrolle
 - Aufgabenbereich, 25
- Kurzname, 85
- label, 85
- Label, 115
 - Bandlabel, 51, 85, 87, 89, 96, 115
 - Filelabel, 51, 72, 89, 115
 - Medienlabel, 24, 53, 96, 100, 115
- Letter, 81
- limit, 90
- Linux, 115
- list, 93
- Logging, *siehe* Sicherung
- Mail, *siehe* Datentyp
- MByte, 115
- modulo, 89
- mSQL, 66, 68, 109–111
- mysql-cron-dump, 67
- Multiplex, 20, 28–30

- Name*, 82
- News, *siehe* Datentyp, Usenet
- nick, 85
- no_bsf, 97
- no_eom, 97
- nolabel, 86
- none, 93
- notify, 92
- Number*, 81

- Objektnamen, 79, 83
- Operator, 17, 24, 70, 115
 - interface, 17, 115
- operator interface, *siehe* Operatorinterface
- output, 87
- owner, 88

- Pathname*, 82
- Perl-5, 63, 110, 120
- PhysicalSizes*, 98
- Pipe, 79, 83, 87, 116
- POSIX, 119, 120
- Prefix, 60
- Priorität, 37–39, 74
 - Anfangspriorität, 37, 72
- Prozeß, 116

- quit, 93
- QuotedString*, 82

- readonly, 96
- release, 91
- ReleaseOptions*, 91
- Request for Comments, *siehe* RFC
- Resource*, 27, 79–103, 116
- Restaurierung, 24, 47
 - Aufgabenbereich, 25
 - Fragebogen, 105
- Restaurierungsprogramm, 43, 49, 50
- restore, 56
- restrict, 94
- RFC, 116, 120

- rootonly, 95

- Schwellwert, 11–14
- scsiblk, 98
- scsidensity, 98
- Separator*, 94
- Sequenz, 9, 11–14
- Server, 17, 116
- setinput, 88
- SetOptions*, 89
- setoutput, 88
- SGML, 116
- Sicherung
 - Gesamtsicherung, 8–13
 - inkrementelle, 8–12
 - Intervall, *siehe* Intervall
 - Logging, 12
 - Sequenz, *siehe* Sequenz
 - Zeitpunkt, 1, 32–37, 71
- Sicherungsprogramm, 40, 43, 50, 77

- Size*, 90
- slot, 86
- slots, 96
- smart*, 23, 116
 - Benutzerschnittstelle, 44
- Solaris, 66, 116
- Standardized Generalized Markup Language, *siehe* SGML
- Statistik, 108
 - Änderungshäufigkeit, 6, 13, 58, 107
 - Aufgabenbereich, 25
 - Betriebsstatistik, 13, 18, 108
- status, 92
- Steuerung
 - Aufgabenbereich, 25
- SunOS, 66, 116
- Syntaxdiagramm, 80–99
- SysCmd*, 83
- System, *siehe* Datentyp
- System V, *siehe* SYSV
- SYSV, 66, 116

Task, 64
telnet, 85, 120
Timeout, 41
transport, 90
Transport
 Aufgabenbereich, 24
 TransportOptions, 90

Uniform Resource Locator, *siehe*
 URL
UNIX, 117
URL, 117, 120
Usenet, 6, 117
user interface, *siehe* Benutzerin-
 terface

Verbinden, 38, 74
Verwaltung
 Aufgabenbereich, 24
Verzögerungsfaktor, 37, 72
VM, 27–31, 63, 77, 109, 110, 117
VM-Info, 41
Volume, 17, 117
 -ID, *siehe* Volumenummer
 -nummer, 17, 49, 52, 74, 117
 Manager, *siehe* VM

World Wide Web, *siehe* WWW
WWW, 44, 111, 117

xdmui, 26, 100

Zeitrahmen, 32–36, 70, 73
Zugriffsrecht, *siehe* ACL