DATENSICHERUNG

Nur zum internen Gebrauch und für Schulungszwecke.

Unbereinigte Vorabversion.

TA TRIUMPH-ADLER AG

Nürnberg

Änderungen vorbehalten
Juli 1987

Vorgesehen für Änderungsdienst Datensicherung

Seite Bemerkung

INHALTSVERZEICHNIS

Auflistung der ABBILDUNGEN

18. Datensicherung

18.1 Allgemeines

Unter Datensicherung und Recovery versteht man Maßnahmen zur Verhinderung von Datenverlusten, wie sie durch Hardware- und Software-Fehler auftreten können.

Hierzu gehört die periodische Erstellung von Kopien von relevanten Datenbeständen auf externen Datenträgern.

Allgemein verbreitet gegen größere Datenverluste und für die Datenrestaurierung ist die Anwendung des Generationenprinzips und insbesondere des "Drei-Generationen-Prinzips":
Bei Fortschreiben des Datenbestandes werden jeweils 3 Generationen der Sicherungskopien auf Datenträger archiviert, damit auch bei Verlust einer Sicherungskopie die Datenrestaurierung möglich ist.

Unter Datensicherung im Rahmen dieser Beschreibung verstehen wir Einrichtungen, die das Kopieren von konsistenten Datenbeständen und das Laden solcher Datenbestände unterstützen.

Dazu gehören sowohl Softwarewerkzeuge als auch organisatorische Richtlinien und Maßnahmen.

In dieser Beschreibung wird auf die Maßnahmen eingegangen, die in System M32-Konfigurationen die Datensicherung unterstützen. Unterstützt werden neben zentralen Datenbeständen auf "shared logic" Systemen auch verteilte Datenbestände auf "shared resources" Systemen (Verbundsysteme, Netzwerke).
Eine verteilte Datenhaltung, also die Speicherung eines logisch zusammengehörigen Datenbestandes auf mehreren Plattenspeichern verschiedener Rechner, sollte nur in Ausnahmefällen in Betracht gezogen werden. Wegen der Komplexität und Störungsanfälligkeit der Datensicherung in derartigen Konfigurationen sollte verteilte Datenhaltung so weit wie möglich vermieden werden.

Diese Dokumentation ist als Beschreibung für Systemanalytiker und Systemprogrammierer gedacht. Beschrieben werden das Datensicherungskonzept und die Softwaretools, die für die Realisierung einer kunden- bzw. anwendungspezifischen Datensicherung zur Verfügung stehen.

Es wird darauf hingewiesen, daß für die Datensicherung in konkreten Installationen ein installationsspezifisches Verfahren festzulegen und eine darauf abgestimmte Benutzerdokumentation zu erstellen ist.

18.2 Sicherungsarten

Bei der Datensicherung unterscheidet man zwischen logischer und physikalischer Datensicherung.

Unter einer physikalischen Datensicherung versteht man die sektor- oder spurweise Kopie des Mediums, unabhängig von Daten- und Verwaltungsstrukturen. Bei System M32 können gesamte Platten oder auch Plattenbereiche physikalisch auf Band kopiert werden, die internen Verwaltungsdaten und die physikalische Datenstruktur einer Platte oder Teilplatte bleiben dabei unverändert.
Um möglichst geringe Kopierzeiten zu erreichen, erfolgt der Kopiervorgang im "Streaming Mode", d.h. physikalisches Lesen von Plattensektoren und direktes Aufzeichnen auf Band mit Hardware-Unterstützung durch den Platten/Streamer Controller unabhängig vom Betriebssystem.

Bei der logischen Datensicherung können Dateien oder Directories einzeln gesichert werden. Die Daten werden dabei satz- oder blockweise unter Verwendung der I/O-Mechanismen des Betriebssystems kopiert. Es können physikalische und z.T. auch logische Fehler beim Sichern erkannt werden. Wegen der satz-/blockweisen Verarbeitung sind die Kopierzeiten jedoch länger als bei der physikalischen Sicherung.

18.3 Recovery

Unter Recovery in diesem Zusammenhang versteht man die Wiederherstellung von korrekten konsistenten Datenbeständen nach einer Fehlersituation. Ein Datenbestand ist immer dann konsistent, wenn alle Transaktionen (siehe Anmerkung), die auf einen Datenbestand zugreifen, vollständig durchgeführt sind.

Anmerkung
Eine Transaktion stellt einen in sich abgeschlossenen Benutzerauftrag dar. Dabei wird in diesem Zusammenhang vorausgesetzt, daß ein solcher Benutzerauftrag, wenn er für sich allein abläuft, korrekt abläuft, so daß also die Datenbank (hier Datenbestand) durch eine Transaktion von einem konsistenten wieder in einen konsistenten Zustand überführt wird.
[G.Schlageter/W.Stucky: Datenbanksysteme: Konzepte und Modelle - Teubner-Verlag 1983]
Konsistent ist ein Datenbestand, wenn seine Daten in Bezug auf die jeweilige Anwendung widerspruchsfrei sind. Für die Konsistenz eines Datenbestandes sind nur verändernde Zugriffe von Bedeutung.

Ein Datenbestand ist - mit großer Wahrscheinlichkeit - während der Durchführung von Teilaufträgen einer Transaktion auf einen Datenbestand inkonsistent. Tritt in einem solchen Zeitpunkt eine Fehlersituation auf, ist der Datenbestand inkonsistent und für die Wiederherstellung eines konsistenten Datenbestandes muß Recovery durchgeführt werden.

Fehlersituationen können sein:

- Programmabbruch, -deadlock
- Fehler im Dateisystem oder Betriebssystem
- Fehler in CPU oder Hauptspeicher
- Unlesbarkeit der Daten auf Plattenspeicher.

Transaktionskonzept, Recoveryverfahren und Logbuchführung (siehe Anmerkung) sind ausführlich in der Datenbankliteratur beschrieben. Diese Verfahren gelten auch für Dateisysteme. Im Gegensatz zu Datenbanksystemen gibt es jedoch i.a. in Dateisystemen keine Systemunterstützung für Transaktionssteuerung und Logbuchführung. Die erforderlichen Verfahren sind Bestandteile der Anwendungsprogramme und sie müssen daher bereits bei der Konzeption der Anwendungsprogramme berücksichtigt werden. Im Rahmen der hier beschriebenen Datensicherung wird nur das in der Literatur als Langzeitrecovery oder Rekonstruktion bezeichnete Verfahren betrachtet:

1. Laden einer Kopie des (konsistenten) Datenbestandes, die zu einem früheren Zeitpunkt hergestellt wurde.
2. Wiederholung aller seit diesem Zeitpunkt auf den Datenbestand durchgeführten Datenmanipulationen.

Die Kopie eines Datenbestandes wird durch die Datensicherung erstellt, wobei während der Datensicherung gewährleistet sein muß, daß der Datenbestand korrekt und konsistent bleibt. Die erforderlichen Maßnahmen für die Wiederholung aller seit diesem Zeitpunkt durchgeführten Datenmanipulationen sind Bestandteil der Anwendungssoftwarepakete. Wichtig ist die exakte Synchronisation des Zeitpunktes der Datensicherung und Beginn der Wiederholung der durchgeführten Datenänderungen. Die Verfahren für Wiederholung von Datenmanipulationen auf den Datenbestand unterscheiden sich nach

- Stapelverarbeitung und
- Dialogverarbeitung.

Anmerkung
Unter Logbuchführung wird die Aufzeichnung sämtlicher Verfahrensschritte einer Transaktion verstanden.

Bei Stapelverarbeitung genügt meist eine Wiederholung des Stapellaufes während bei Dialogverarbeitung aufwendige Verfahren z.B. Logbuchführung notwendig sind. Werden von den Anwendungsprogrammen keine Vorkehrungen für die automatische Wiederholung geschaffen, so müssen alle Transaktionen manuell nachvollzogen werden.

Besondere Schwierigkeiten treten bei Recovery auf, wenn mehrere logische Datenbestände auf einer Platteneinheit gespeichert sind, und physikalische Datensicherung verwendet wird. Bei einem Fehler in einem Datenbestand muß dann auch für den anderen Datenbestand Recovery durchgeführt werden.

18.4 Relevante Konfigurationen für Datensicherung

Vor jeder Datensicherung ist zu prüfen, ob die Sicherung durchgeführt werden darf.

Es ist dabei sicherzustellen, daß die betroffenen Dateien zum Sicherungszeitpunkt keinem modifizierenden Zugriff unterliegen. Jeder weitere Zugriff muß für die Dauer der Sicherung unterbunden werden.

Dies kann nicht allein durch die Sicherungswerkzeuge geschehen.

Zulässigkeitsprüfungen dieser Art sind abhängig von der vorliegenden Rechnerkonfiguration zu gestalten. Die Komplexität nimmt mit der Konstellation der teilnehmenden Systeme zu.

18.4.1 Single User System

Konfigurationsmerkmale:

- nur ein Benutzer arbeitet an diesem System
- relevante Datenbestände sind auf der lokalen Platte abgespeichert

Der Benutzer kann die von ihm gestarteten Anwendungsprogramme vollständig überwachen. Er ist in der Lage, alle Zugriffe auf relevante Datenbestände zu kontrollieren. (Abbildung 18-1)

Hier kann die Zulässigkeitsprüfung einfach gehalten werden. Der Benutzer, der eine Sicherung vornehmen will, kann im Regelfall selbst sicherstellen, daß kein Zugriff auf die zu sichernden Daten erfolgt.

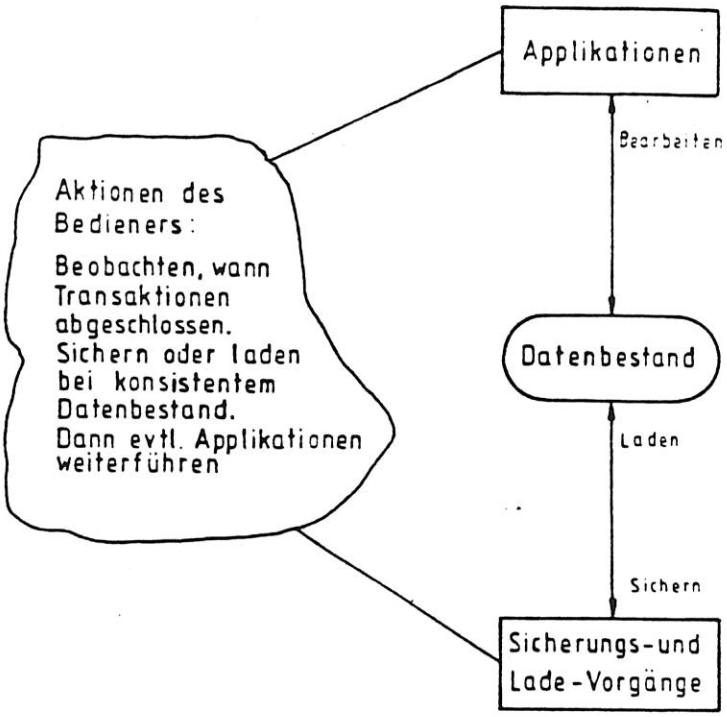

Abbildung 18-1: Single User System

18.4.2 Multi User System mit zentralem Datenbestand

Konfigurationsmerkmale:

- an diesem System können mehrere Benutzer arbeiten
- relevante Datenbestände sind lokal auf der Platte abgespeichert.

Wie in Abbildung 18-2 dargestellt, bedeutet das, daß mehrere Benutzer gleichzeitig den gleichen Datenbestand bearbeiten können.
Es können verschiedene Anwendungspakete parallel ablaufen bzw. ein Anwendungsprogramm kann auch mehrfach (durch verschiedene Benutzer gestartet) ablaufen.

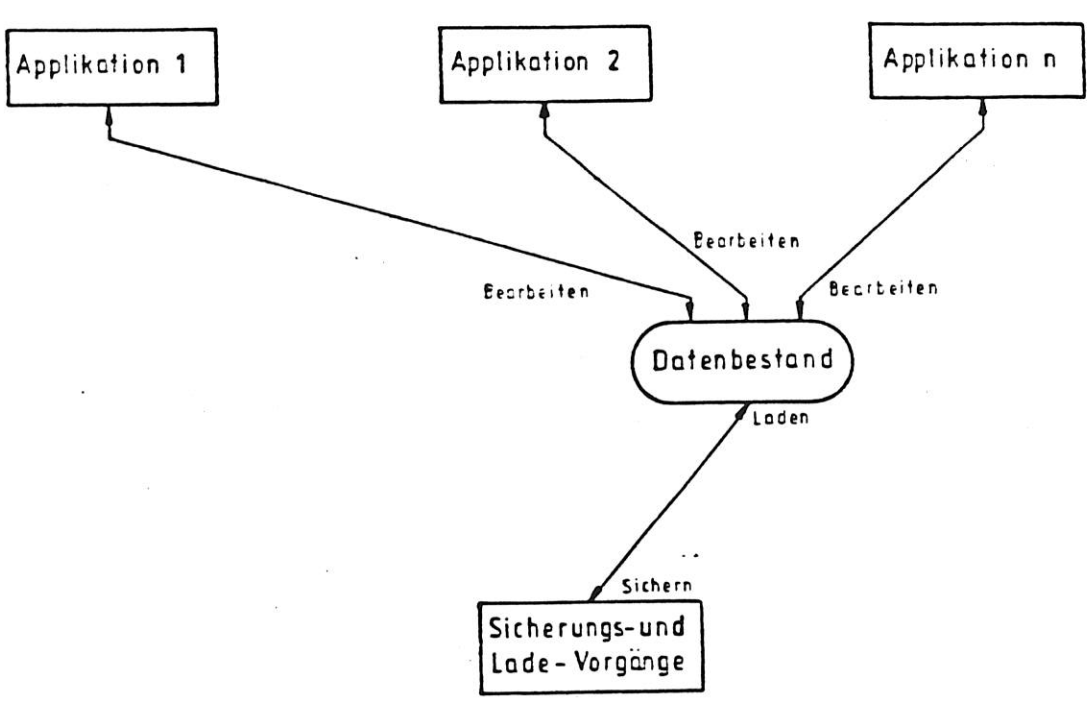

Abbildung 18-2: Multi-User-System mit zentralem Datenbestand

Hier ist die Zulässigkeitsprüfungmit Unterstützung der Anwendungsprogramme vorzunehmen.

Jedes Anwendungsprogramm muß vor der Bearbeitung von Daten zunächst den Zugriff prüfen und sich bei vorliegender Berechtigung dem Gesamtsystem als Benutzer anmelden.

Nach erfolgter Manipulation der Dateien bzw. bei Abschluß der Anwendung muß dementsprechend die Abmeldung erfolgen.

Die Sicherungsverfahren können nun anhand der angemeldeten Benutzer überprüfen, ob eine Sicherung zulässig ist.

Vor Durchführung der Sicherung ist der Datenbestand zu sperren, und damit kein unerlaubter Zugriff durch Anwendungsprogramme während der Sicherung möglich ist.

Diese Konfiguration ist für Multi User Systeme üblich. Eine zentrale Datenhaltung sollte, wenn möglich, immer verwendet werden. Falls aus

Speicherplatzgründen eine verteilte Datenhaltung erwogen wird, muß in besonderem Maße berücksichtigt werden, daß die Datensicherung aufwendig sein kann.

Solche Konfigurationen können in "shared logic" oder "shared resources" Systemarchitekturen realisiert werden.

18.4.3 Multi User System mit dezentralem Datenbestand

Konfigurationsmerkmale:

- Mehrere Benutzer arbeiten an einem System
- mehrere Systeme sind über LAN gekoppelt
- Der logisch zusammenhängende Datenbestand ist verteilt auf mehrere Platten, die auf unterschiedlichen Netzknoten installiert sind.

Wie Abbildung 18-3 darstellt, müssen deshalb die einzelnden Teildatenbestände separat an den einzelnen Knoten gesichert werden.

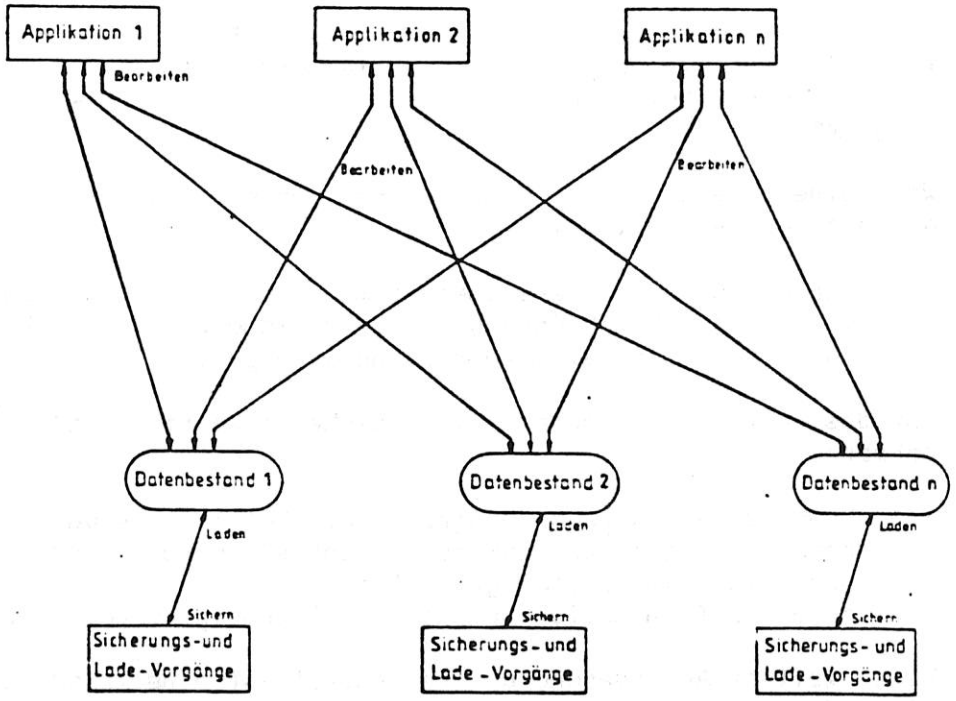

Abbildung 18-3: Multi-User-System mit dezentralem Datenbestand

Bei physikalischer Sicherung muß die Datensicherung lokal durchgeführt werden (während des Streaming-Modes ist hier kein Netzbetrieb möglich). Aber auch bei logischer Sicherung ist aus Durchsatz- und Geschwindigkeitsgründen die lokale Sicherung zu empfehlen.

Hier ist die Zulässigkeitsprüfung ebenfalls mit Unterstützung der Anwendungsprogramme vorzunehmen. Außerdem muß zusätzlich organisatorisch der Zeitpunkt der Sicherung festgelegt werden.

Weiterhin muß möglich sein, netzweit die zusammengehörenden Datenbestände einerseits für die Anwendungen als zugreifbar zu kennzeichnen und andererseits zu Beginn eines Sicherunglaufes zu sperren.

Der organisierte Wechsel zwischen Anwendungsphase und Sicherungsphase wird am einfachsten von einem festgelegten Netzknoten aus gesteuert.

Die eigentliche Sicherung hat an dem Knoten, an denen die Daten residieren, selbst zu erfolgen.

18.5 Sicherungskonzept

Das im folgenden beschriebene Verfahren unterstützt die Datensicherung für Multiuseranwendungen mit zentralem und verteiltem Datenbestand. Für Single-User-Systeme sind keine besonderen Absicherungen der Datensicherung erforderlich.

Wenn eine Datensicherung durchgeführt werden soll, müssen folgende Voraussetzungen erfüllt sein:

- Vor Beginn der Datensicherung muß der Datenbestand konsistent sein.
- Während der Durchführung der Datensicherung dürfen keine Datenmanipulationen auf dem Datenbestand erfolgen.

Für das im folgenden beschriebene Sicherungskonzept gelten folgende Annahmen:

- Wenn keine Anwendungsprogramme auf Dateien des Datenbestandes (verändernd) zugreifen, ist der Datenbestand konsistent.
- Wenn ein Anwendungsprogramm beendet wird, sind alle Aufträge, die zu einer Transaktion gehören, vollständig durchgeführt.

Eine weitere Anforderung an das Konzept ist eine kompatible Lösung aus Sicht der Anwendungsprogramme für Konfigurationen mit zentralem Datenbe-

stand und Konfigurationen mit verteilten Datenbeständen.

Die Grundlage des Sicherungskonzeptes ist ein Synchronisationsmechanismus für den gegenseitigen Ausschluß von Zugriffen aus Anwendungsprogrammen auf den Datenbestand und dem Kopiervorgang während der Datensicherung, d.h. wenn Benutzerprogramme aktiv sind, kann keine Datensicherung durchgeführt werden und umgekehrt, wenn Datensicherung abläuft, dürfen keine Benutzerprogramme auf den Datenbestand zugreifen.
Die zentrale Datenstruktur im DASI-System für die Synchronisation besteht im wesentlichen aus zwei Einträgen:

1. act_user:
   Anzahl der z.Zt. aktiven Benutzer eines Datenbestandes
2. ds_mode
   Aktueller Zustand des Datenbestandes
   Wichtige Zustände sind:

USMOD:	Benutzermodus
SAVREQ:	Sicherungswunsch angemeldet
SAVMOD:	Sicherungsmodus
SL_END:	Ende der Datensicherung

In der Dateistruktur sind zusätzliche Informationen gespeichert:
- Verwaltungsinformationen
- Benutzereinträge für Auskunftsfunktionen
- Kennung für Datenbestand
- Versionsnummer für Datensicherung

18.5.1. Vorgehensweise

Aus Anwendungsprogrammen / -paketen

1. Jeder Benutzer (Programm), der auf den Datenbestand zugreifen will, meldet sich vor dem ersten Zugriff im DASI-System an. Befindet sich der Datenbestand im "USMOD" darf der Anwender uneingeschränkt auf den Datenbestand zugreifen. Befindet sich der Datenbestand nicht im "USMOD" wird die Anmeldung abgelehnt.
   Funktion: "tstaent"

2. Bei Programmende nach dem letzten Zugriff auf den Datenbestand meldet sich der Benutzer im DASI-System ab.
   Funktion: "leavesys".

Aus Sicherungsprozedur

1. Vor Beginn des Sicherungslaufes wird der Sicherungswunsch im DASI-System angemeldet. Wenn keine aktiven Benutzer im DASI-System angemeldet sind, wird der Zustand des Datenbestandes auf "SAVMOD" gesetzt und die Datensicherung kann durchgeführt werden. Wenn aktive Benutzer im DASI-System angemeldet sind, wird der Zustand auf "SAVREQ" gesetzt und die Datensicherung wird abgelehnt. Weitere Benutzer werden dann nicht mehr zugelassen. Aktive Benutzer in Anwendungspaketen werden jedoch nicht unterbrochen.Der Aufruf muß zu einem späteren Zeitpunkt wiederholt werden.
   Kommando: "testalock".

2. Nach Beendigung des Sicherungslaufes wird Zustand des Datenbestandes auf "USMOD" zurückgesetzt
   Kommando: "resetlock".

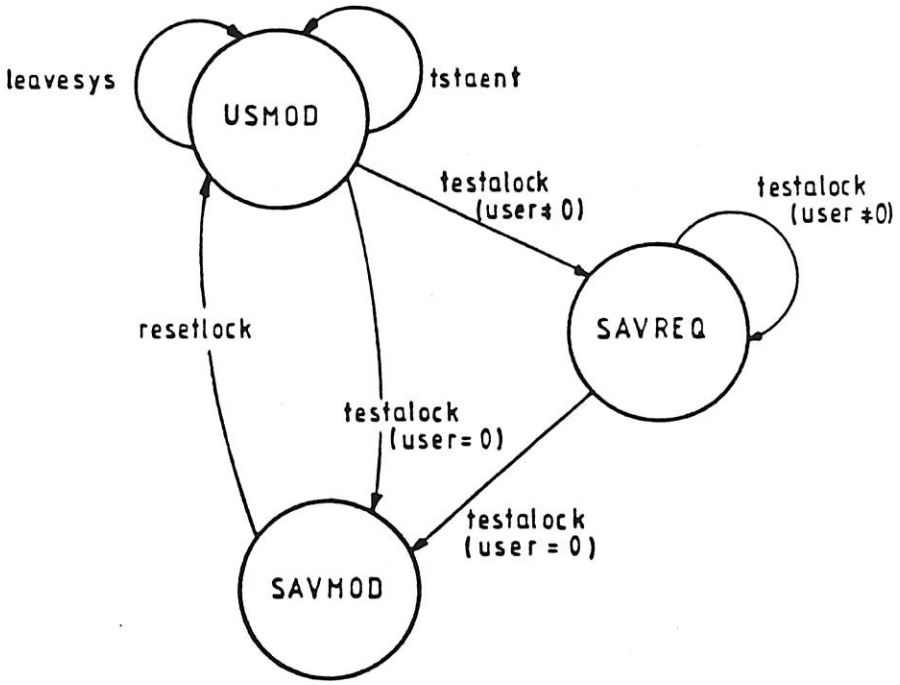

Abbildung 18-4: Zustandsdiagramm: Zentraler Datenbestand

Der bisher beschriebene Ablauf gilt für Konfigurationen mit zentralem Datenbestand. Bei verteiltem Datenbestand gibt es zusätzlich den Zustand

"SL_END" Ende der Datensicherng und zwei weitere Kommandos:

sminit: Initialisieren Datensicherung
smexit: Ende Datensicherung = Initialisieren Benutzermodus

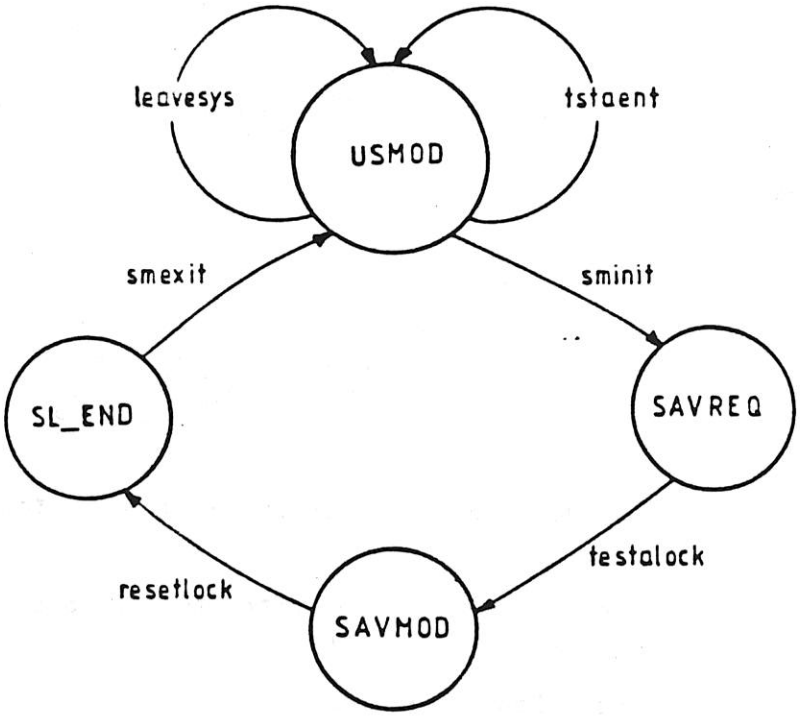

Abbildung 18-5: Zustandsdiagramm: Verteilter Datenbestand

18.5.2 Konfiguration mit zentralem Datenbestand

In Abbildung 18-6 sind alle für die Synchronisation von Anwendungsprogrammen und Sicherungsprogrammen bei Multiusersystemen mit zentralem Datenbestand erforderlichen Instanzen und Funktionen dargestellt.
Die zentrale Datenstruktur für die DASI-Steuerung ist in dem "Control-File" gespeichert. Die Synchronisation der Anwendungsprogramme und dem Sicherungslauf erfolgt

- aus den Anwendungsprogrammen mit den Funktionen "tstaent" und "leavesys".
- aus den Sicherungs- bzw. Ladeprogrammen mit den Kommandos "testalock" und "resetlock".

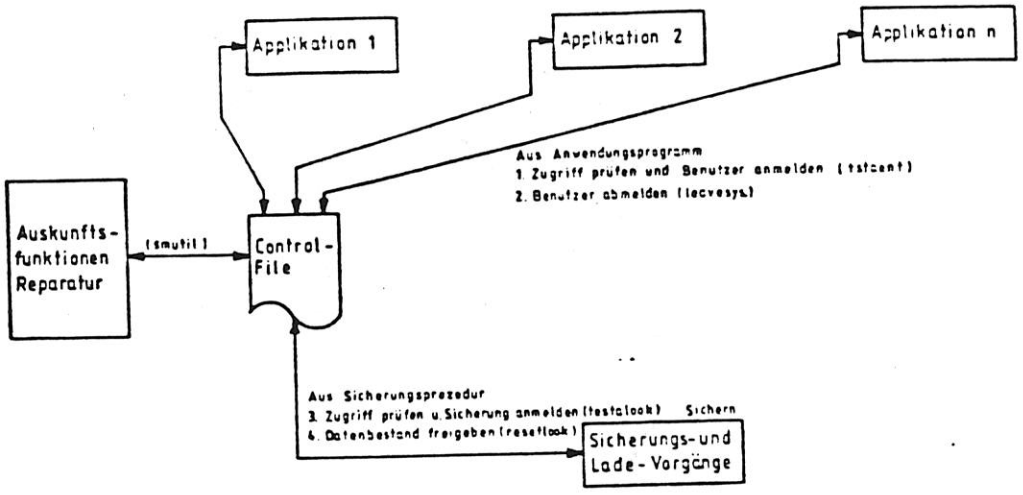

Abbildung 18-6: Steuerdateien bei zentralem Datenbestand

Für die Administration der DASI Datenstruktur ist das Kommando "smutil" verfügbar. Es enthält Funktionen für

- Kreieren eines "Control-Files"
- Modifizieren der Parameter im "Control-File" (erforderlich bei inkonsistentem Control-File)
- Information über aktive Benutzer des Datenbestandes. Angezeigt werden:
  Login-Name, Stationsname und tty-name

18.5.3 Konfiguration mit dezentralem Datenbestand

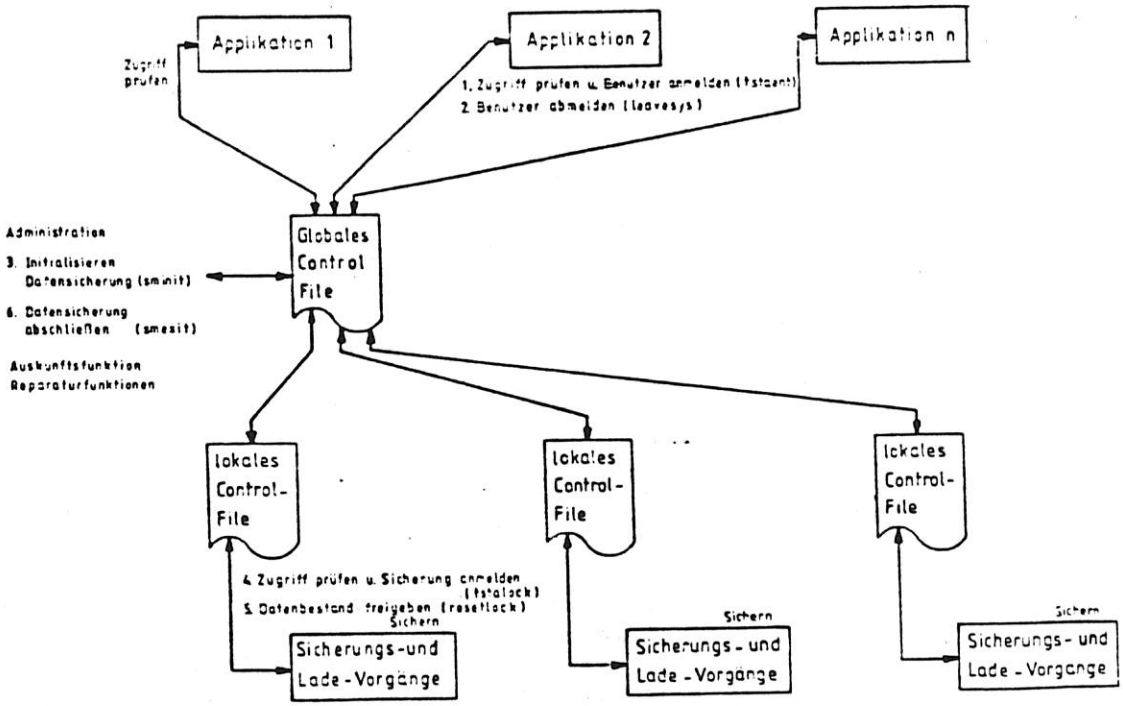

Abbildung 18-7: Steuerdateien bei dezentralem Datenbestand

Bei Konfigurationen mit dezentralem Datenbestand ist ein logischer Datenbestand unterteilt in auf verschiedenen Netzknoten installierten Teildatenbeständen. Die Abbildung 18-7 zeigt alle für diese Anwendung erforderlichen Instanzen und Funktionsaufrufe. Neben dem "Global Control File", in dem wie bei der Konfiguration mit zentralem Datenbestand die Anwendungsprogramme sich an- bzw. abmelden, existiert für jeden Teildatenbestand ein "Local Control File". Das "Local Control File" muß auf dem gleichen Netzknoten installiert sein wie der Teildatenbestand und dient zur Überwachung und Steuerung für die Sicherung dieses Teildatenbestandes. Die Funktionsaufrufe aus Anwendungsprogrammen und Sicherungsprozeduren sind gleich wie bei Verfahren mit zentralem Datenbestand. Zusätzlich sind folgende Funktionen erforderlich:

sminit: Initialisiere Datensicherung
Das Kommando prüft, ob Datensicherung zulässig ist.

Für "aktive User" gleich 0 gilt:
Status Datenbestand im "Global Control File" wird auf "SAVMOD" gesetzt, d.h. der Datenbestand wird für die Benutzer gesperrt. Die Zustände aller Teildatenbestände werden in den Local Control Files auf SAVREQ gesetzt, d.h. die Teildatenbestände können und müssen anschließend gesichert werden.

Für "aktive User" ungleich 0 gilt:
Der Status im "Global Control File" wird auf SAVREQ gesetzt, d.h. weitere Anmeldungen aus Anwendungsprogrammen werden abgelehnt. Zulässig sind nur Abmeldungen. Der Auftrag für Initialisierung der Datensicherung wird abgelehnt. Sminit muß zu einem späteren Zeitpunkt wiederholt werden.

smexit: Datensicherung abschließen
Das Kommando prüft, ob alle Teildatenbestände gesichert wurden.

Für Sicherung durchgeführt gilt:
Die "Local Control Files" werden auf "USMOD" gesetzt, d.h. es darf danach keine Datensicherung durchgeführt werden und das "Global Control File" wird auf "USMOD" gesetzt, d.h. der gesamte logische Datenbestand steht uneingeschränkt für Benutzerzugriffe zur Verfügung.

Für Sicherung nicht, bzw. nicht vollständig durchgeführt gilt:
Eine Meldung wird ausgegeben. Das Kommando muß zu einem späteren Zeitpunkt wiederholt werden.

In den Abbbildungen 18-8 und 18-8 sind die Zugriffe auf den Datenbestand und die Funktionsaufrufe des Datensicherungssystems für die aus Sicht der Datensicherung relevanten Konfigurationen dargestellt.

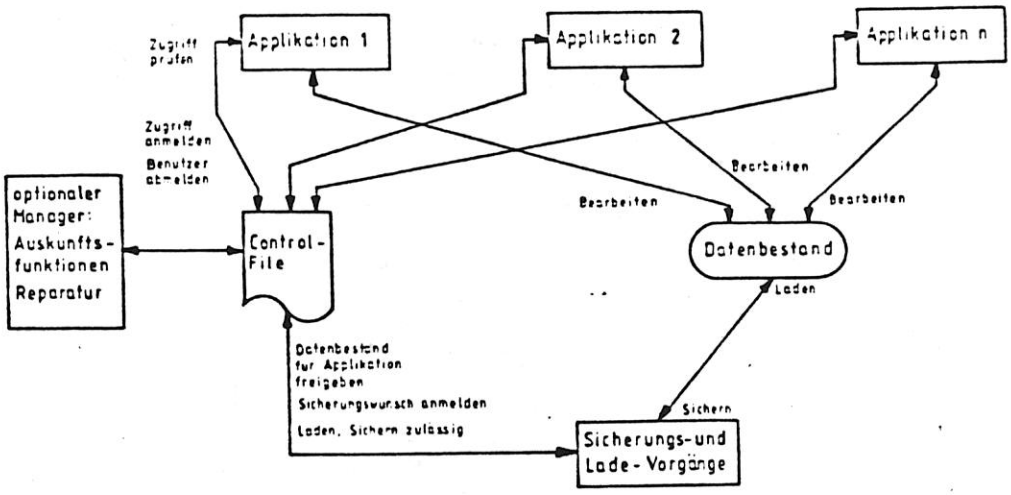

Abbildung 18-8: Sicherungsaktionen bei zentralem Datenbestand

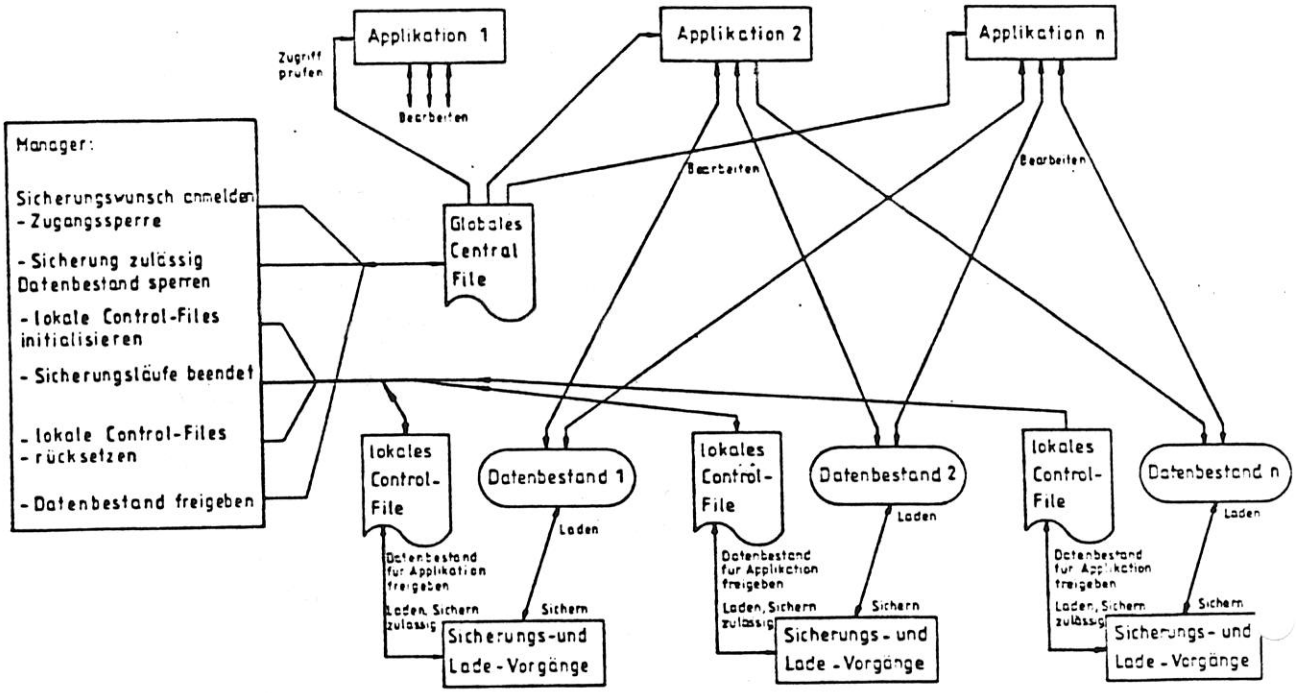

Abbildung 18-9: Sicherungsaktionen bei dezentralem Datenbestand

18.6 Sicherungswerkzeuge

Für die Datensicherung und das hier beschriebene Konzept werden Bibliotheksfunktionen und Kommandos zur Verfügung gestellt, die dem Systemanalytiker und Systeminstallateur ermöglichen, mit geringem Aufwand kundenspezifische Datensicherungsverfahren zu definieren und zu realisieren. Voraussetzung ist jedoch, daß die Anwendungsprogramme, -pakete sich im Datensicherungssystem an- bzw. abmelden.
Die Sicherungswerkzeuge werden wie folgt unterteilt:

1. Aufrufe aus Anwendungsprogrammen, -paketen
   - testaent
   - leavesys

   Diese Funktionen sind Bibliotheksfunktionen und müssen zu den Anwendungsprogrammen oder -paketen gebunden werden. Parameterversorgung und Verarbeitung der Rückgabewerte erfolgt im Anwenderprogramm

2. Kommandos für Sicherungs- bzw. Ladeprozeduren
- testalock
- resetlock

3. Kommandos für Initialisierung und Abschluß der Datensicherung
- sminit
- smexit

4. Kommando für Datensicherungsadministration
- smutil

Ausführliche Beschreibung der Sicherungswerkzeuge befinden sich im Anhang dieser Dokumentation.

18.7 Aufrufe aus Anwendungspaketen bzw. -programmen BBP/Kommerz

18.7.1 Kommerzielle Anwendungspakete

Für M-Kommerz und alle anderen Anwendungspakete, die unter dem TA-Ablaufsystem TAKOS installiert sind, erfolgt der Aufruf von 'tstaent' bzw. 'leavesys' bei Initialisierung bzw. Beendigung des Anwendungspaketes. Welche und ggf. wieviele Controlfiles hierbei angesprochen werden sollen, ist parametrisierbar und wird beim Einrichten des Systems vom Kundenorganisator entsprechend der spezifischen Kundenkonfiguration festgelegt.

Bei fremden Paketen (floating software), die eine eigene Ablauf- und Menü-Steuerung haben, z.B. ein Cobol-Rahmenprogramm oder über Shell-Scripts, muß der Software-Ersteller die Positionierung der Aufrufe und ggf. die Parametrisierung selbst festlegen. Es muß dann also aus Cobol heraus ein CALL-Aufruf abgesetzt bzw. aus Shell ein kleines C- oder Cobolprogramm aktiviert weren, das diesen Aufruf beinhaltet.

Ein Beispiel für den Aufruf aus einem C-Proramm und die entsprechende Statusbehandlung finden Sie im Anhang Programmbeispiel für Testzwecke, Abschnitt 18.11.4.

18.7.2 Büro-Basis-Programme

Jeder Benutzer hat unter TABOS seinen privaten Datenbestand und hat eventuell Zugriff auf Archivdaten. Der private Datenbestand liegt unter einem TANIX-Dateisystem. Der Name dieses Dateisystems wird in der Shell-Variablen "DTEOBJ" gehalten.

Die Verwaltung aller Daten unter diesem Dateisystem geschieht durch den "Elektron Schreibtisch".

Im Programm des "Elektron Schreibtisch" wird beim Start in der Datei "$DTEOBJ/dasi.config" nachgesehen, ob der "Elektron Schreibtisch" sich zu Beginn in einem Control-File eintragen muß. Wenn mehrere Einträge in der Datei "dasi.config" vorhanden sind, erfolgt in jeder dieser Control-Files ein Eintrag mit "tstaent".

Beim "Archiv" wird auf Serverseite eine Datei "$HOME/bin/data/dasi.config" unter dem Benutzer "dtarchiv" dafür geführt. Das Eintragen erfolgt analog der Vorgehensweise beim "Elektron Schreibtisch".

18.8 Ablauf der Datensicherung

18.8.1 Allgemeines

Für den komfortablen Aufruf einer begrenzten Auswahl an Kommandos steht das Utility "das" zur Verfügung.

Dieses Menüprogramm verwendet eine Steuerdatei, in der das Menübild, die Anwahlnummern und die korrespondierenden Kommandos festgelegt sind. Diese Datei kann, unter Beachtung der festgelegten Syntax, beliebig editiert und damit den speziellen Anforderungen angepaßt werden. (Genaue Beschreibung im Anhang, 18.11.2.2)

Mit Hilfe dieses Programmes kann der Ablauf einer Datensicherung so aufgebaut werden, daß sich sowohl der save manager als auch ein mit der lokalen Sicherung betrauter Benutzer an seiner Station unter dem Login-Namen "dasi" anmeldet. Dann wird automatisch das Menüprogramm gestartet und es kann ausgewählt werden, welcher Datenbestand gesichert bzw. geladen werden soll, z. B. Elektron Schreibtisch, Archiv, Kommerz A, Kommerz B, usw..

Beim Entwurf der Sicherungsprozeduren müssen sorgfältige Überlegungen angestellt werden, um eine Konsistenz der gesicherten Daten zu gewähr-

leisten. Das gilt besonders bei physikalischer Sicherung, wenn Datenbereiche, die logisch verschiedenen Anwendungspaketen angehören, unter dem gleichen Filesystem angelegt wurden. Beispiele und Empfehlungen zu dieser Problematik befinden sich im Anhang unter Musterprozeduren, Abschnitt 18.11.5.

18.8.2 Datensicherung für kommerzielle Anwendungen

Die Initialisierung der Datensicherung darf nur von einer bestimmten ausgezeichneten Station (Systemadministrator) nach Überprüfung der Zugriffsberechtigung gestartet werden. Während dieser Zeit kann keine neue kommerzielle Anwendung mehr angemeldet werden. Nach erfolgreicher Initialisierung kann die eigentliche Datensicherung ablaufen.

a) SL-Systeme und SR-Systeme mit zentralem Datenbestand:
Die Datensicherungsprozedur wird zentral aufgerufen und ausgeführt.

b) SR-Systeme mit verteiltem Datenbestand:
Die Datensicherungsprozedur wird lokal aufgerufen und ausgeführt.

Die Datensicherung für kommerzielle Anwendunen wird vorzugsweise auf der physikalischen Ebene ablaufen, bei Anlagen ohne Streamer wird jedoch auch die logische Datensicherung über Diskette nötig sein.

Nach Beendigung des lokalen Datensicherungslaufes wird die über 'resetlock' ausgegebene aktuelle Versionsnummer manuell auf den Datenträger übertragen.

Sind alle lokalen Datensicherungsläufe erfolgreich beendet, wird die Datensicherung durch den Systemadministrator an der ausgezeichneten Station abgeschlossen und für die kommerzielle Verarbeitung freigegeben.

18.8.3 Sicherungsprogramm "Elektron Schreibtisch"

Durch die Sicherung eines "Elektron Schreibtisch" werden die Verwaltungsdaten und alle privaten BBF-Daten (außer Archiv-Daten) eines Benutzers (alle Daten, die unter dem Dateisystem "$DTEOBJ2 liegen) gesichert.

Jeder "Elektron Schreibtisch" kann von jeder Station im Netz aus gesichert werden.

Ablauf:

- Eingabe des Benutzernamens, dessen "Elektron Schreibtisch" gesichert werden soll.

- Eingabe des Stationsnamens, auf dem der Benutzer arbeitet.

- Eingabe des Paßwortes des Benutzers.
  Das Paßwort wird mit dem Paßwort auf der oben bezeichneten Station verglichen. Nachdem überprüft wurde, daß die Daten unter $DTEOBJ auch dem Benutzer gehören und TABOS derzeit nicht aktiv ist, wird der Schreibtisch zum Bearbeiten gesperrt.

- Eingabe, ob auf Streamer oder auf Disketten gesichert werden soll.
  Die Sicherung erfolgt dann mit "lback" auf dem ausgewählten Gerät. Nach der Sicherung wird die Bearbeitung des Schreibtisches wieder erlaubt. Station wird das Directory "$DTEOBJ" bestimmt.

18.8.4 Sicherungsprogramm "Archiv"

Es können nur die Archive gesichert werden, die auf dieser Station liegen.

Wenn physikalisch gesichert werden soll, so können nur alle Archive auf einer Station gemeinsam gesichert werden. Soll physikalisch gesichert werden, darf das Sicherungsprogramm selbst nicht auf einem zu sichernden Dateisystem liegen.

- Falls das Archiv in Control-Files Eintragungen macht, überprüfen, ob in den Control-Files der Sicherungswunsch angemeldet ist.

- Eingabe, ob physikalisch oder logisch gesichert werden soll.

- Falls physikalisch

  - Prüfung, daß alle Archiv-Server passiv sind.

  - Test, ob die zu sichernden Daten auf Dateisystemen liegen, die abgemeldet werden können (Dateisystem darf nicht das root-Dateisystem sein.)

- Im HOME-Directory des Benutzers "dtarchiv" wird eine Datei angelegt, in der alle Dateisysteme eingetragen werden, die gemeinsam gesichert und eingelesen werden müssen.

- Auf jedem physikalischen Dateisystem wird eine Markierungsdatei angelegt, die einem beim Zurückspielen erlaubt, festzustellen, ob das Dateissystem zurückgespielt ist und ob das richtige Dateisystem eingespielt wurde.

- abmelden ("umount") auf alle zu sichernden Dateisysteme.

- Alle Dateisysteme physikalisch sichern. Der Name des Dateisystems ist vom Sicherer auf dem Streamerband zu vermerken. Beim Wiedereinlesen wird er nach dem entsprechenden Band gefragt.

- Alle Dateisysteme wieder anmelden ("mount").

- Markierungsdateien und Datei mit zu sichernden Dateisystemen löschen.

- Falls logisch

  - Eingabe, ob alle Archive auf diese Station oder ein spezielles Archiv gesichert werden soll.

  - Prüfung, daß die zu sichernden Archiv-Server passiv sind.

  - Eingabe, ob die Sicherung auf Streamer oder auf Disketten erfolgen soll.

  - Sicherung aller erforderlichen Directories.

- Falls das Archiv in Control-Files Eintragungen macht (s.o.), dort vermerken, daß die Sicherung durchgeführt wurde.

18.9 Sicherungsprozeduren

Die notwendigen Schritte für die lokale Sicherung am Datenknoten sind in der untenstehenden Übersicht tabellarisch zusammengefaßt. Im Anhang 18.11.5 finden Sie als Beispiel für den Fall der physikalischen und der logischen Datensicherung je ein Shell-Script, das der untenstehenden Tabelle entspricht.

	Bei physikalischer Sicherung	Bei logischer Sicherung
testalock (Datensicherung anmelden)	erforderlich Im Fehlerfall: - Abbruch - Sicherungsanforderung-bleibt bestehen	erforderlich Im Fehlerfall: - Abbruch - Sicherungsanforderung bleibt bestehen
stopnet (Netzbetrieb abmelden)	erforderlich Im Fehlerfall: - Abbruch - Sicherungsanforderung bleibt bestehen	nicht erforderlich
umount (Filesystem abmelden)	erforderlich Im Fehlerfall: - startnet - Abbruch - Sicherungsanforderung bleibt bestehen	nicht zulässig
Datenträger anmelden	erforderlich Im Fehlerfall: - mount - startnet - Abbruch - Sicherungsanforderung bleibt bestehen	evtl. erforderlich Im Fehlerfall: - startnet (falls vorher stopnet) - Abbruch - Sicherungsanforderung bleibt bestehen

	Bei physikalischer Sicherung	Bei logischer Sicherung
Sichern	"dicopy" mit verify Im Fehlerfall: - Datenträger abmelden - mount - startnet - Abbruch - Sicherungsanforderung bleibt bestehen	z. B. "lback" oder "tar" Im Fehlerfall: - evtl. Datentr. abmelden - evtl. startnet - Abbruch - Sicherungsanforderung bleibt bestehen
Datenträger abmelden	erforderlich Im Fehlerfall: - Meldung ausgeben	evtl. erforderlich
resetlock (Datensicherung abmelden)	erforderlich Im Fehlerfall: - Meldung ausgeben Steuerdatei inkonsistent	erforderlich Im Fehlerfall: - Meldung ausgeben Steuerdatei inkonsistent

- Erfolgsmeldung bekannt geben
- Sicherungsversionsnummer bekannt geben (zur Datenträgerbeschriftung), da der Datenträger nicht maschinell gekennzeichnet ist

	Bei physikalischer Sicherung	Bei logischer Sicherung
mount (Filesystem anmelden)	erforderlich Im Fehlerfall: - Meldung ausgeben	nicht erforderlich
startnet (Netzbetrieb anmelden)	erforderlich Im Fehlerfall: - Meldung ausgeben	evtl. erforderlich

18.10 Recovery

18.10.1 Zurücksichern kommerzielle Datenbestände

Bei Systemabsturz oder Programmabbruch sind für M-Kommerz im jeweiligen Bedienerhandbuch, Kapitel 8, Maßnahmen z.T. über Hilfsprogramme beschrieben, die die anwendungslogische Konsistenz der Dateien überprüfen und - soweit möglich - gg. wieder herstellen.
Ist diese Vorgehensweise nicht möglich bzw. nicht ausreichend oder sind aufgrund von Hardware-Fehlern die Dateien nicht mehr lesbar, muß auf der letzten Datensicherung aufgesetzt werden. Es wird hierbei vorausgesetzt, daß nur Anwendungsdateien zerstört sind, Betriebssystem und Programme jedoch weiterhin ohne Einschränkung nutzbar sind, da sonst die Datenrestaurierung durch einen Systemspezialisten vorgenommen werden muß.

Prozedural unterstützt werden lokal die manuell einzugebenden aktuellen Versionsnummern überprüft und die entsprechenden Datensicherungsstände zurückgeladen.

Anschließend müssen alle seit dem letzten Datensicherungszeitpünkt ausgeführten Dialog- und Batchprogramme (Fore-/Backgroundprogramme) in chronologischer Reihenfolge nachgezogen werden.

18.10.2 Zurücksichern "Elektron Schreibtisch"

Jeder "Elektron Schreibtisch" kann von jeder Station im Netz wieder geladen werden.

Ablauf:

- Eingabe des Benutzernamens, dessen "Elektron Schreibtisch" geladen werden soll.

- Eingabe des Stationsnamens, auf dem der Benutzer arbeitet.

- Eingabe des Paßworts des Benutzers.
  Das Paßwort wird auf der oben bezeichneten Station verglichen. Es wird geprüft, bo die Daten unter $DTEOBJ dem Benutzer gehören oder das Directory leer ist.

- Eingabe, ob von Streamer oder Disketten geladen werden soll.
  Nach der Prüfung, daß TABOS nicht gerade aktiv ist, wird der Schreibtisch gesperrt und die bisherigen Schreibtischdaten werden gelöscht. Dann werden die Schreibtisch-Daten vom ausge-

wählten Sicherungsmedium geladen. Abschließend wird die Bearbeitung des Schreibtisches wieder freigegeben.

Der Benutzer muß nun alle seit der letzten Sicherung durchgeführten Änderungen nachziehen.

18.10.3 Zurücksichern "Archiv"

Archive können nur an der Station geladen werden, auf der sie auch liegen. Wenn eine physikalische Sicherung geladen werden soll, darf das Programm zum Laden nicht auf einem zu ladenden Dateisystem liegen.

- Falls das Archiv in Control-Files Eintragungen macht, überprüfen, ob in den Control-Files das Laden erlaubt ist.
- Prüfung, daß alle Archiv-Server passiv sind. Dies muß auch beim Einspielen einer logischen Sicherung gewährleistet werden, da vor dem Laden nicht bekannt ist, ob alle Archive oder nur ein einzelnes gesichert wurde.
- Eingabe, ob eine pyhsikalische oder logische Sicherung geladen werden soll.
- Falls physikalisch
  - "umount" auf des Dateisystem, in dem das HOME-Directory des Benutzers "dtarchiv" liegt.
  - Laden des entsprechenden Dateisystems und "mount".
  - Anhand der beim Sichern angelegten Datei im HOME-Directory des Benutzers "dtarchiv" die restlichen Dateisysteme bestimmen, die noch geladen werden müssen.
  - "umount" auf alle noch zu ladenden Dateisysteme.
  - Alle Dateisysteme laden und "mounten".
  - Überprüfen, ob die richtigen Dateisysteme an den entsprechenden Stellen geladen wurden (durch die beim Sichern angelegten Markierungsdateien).
  - Markierungsdateien und Datei mit zu sichernden Dateisystemen löschen.

- Falls logisch
  - Eingabe, ob vom Streamer oder von Disketten geladen werden soll.
  - Laden der Sicherung.
- Falls das Archiv in Control-Files Eintragungen macht, dort vermerken, daß das (die) Archiv(e) geladen wurde(n).

Alle Archiv-Benutzer müssen nun alle seit der letzten Sicherung durchgeführten Änderungen nachziehen.

18.11 Anhang

18.11.1 Funktionen

18.11.1.1 LEAVESYS(S)

Name:

leavesys - leave system

Definition

```
#include <dasi.h>

leavesys( ctl_file )
char *ctl_file;
```

Parameter

ctl_file Name der Steuerdatei

Beschreibung

Die Funktion "leavesys" überprüft den Zustand und Inhalt der Steuerdatei, bevor sie einen Eintrag des aufrufenden Benutzers entfernt.

Rückgabewert

Bei gültig entferntem Benutzereintrag gibt "leavesys" den Wert 0 zurück.

Bei Fehlern, welche das Betriebssystem entdeckt, wird -1 zurückgeliefert und die Variable errno gesetzt.

In den folgenden Fällen bricht die Funktion ab und gibt eine der in /usr/include/dasi.h definierten Statusmeldungen zurück:

NOACCESS	Kein exklusiver Zugriff auf die Steuerdatei möglich.
ILLCTLFILE	Ungültige Steuerdatei.
NOUSERMOD	Die Steuerdatei ist wider Erwarten nicht im user

mode.

NOENTRY Keine Benutzereinträge vorhanden.

NOTFOUND Benutzereintrag nicht gefunden.

NOPASSWD kein Eintrag in /etc/passwd gefunden.

Dateien

/usr/include/dasi.h enthält die Statusmeldungen
/etc/passwd enthält Login-Namen

Siehe auch

tstaent(S), testalock(C), resetlock(C), sminit(C), smexit(C), smutil(C).

Bemerkung

Die Statusmeldungen NOUSERMOD, NOENTRY, NOTFOUND sind nur möglich, wenn die Steuerdatei inkonsistent geworden ist oder eine falsche Steuerdatei angegeben wurde. Mit "smutil" kann die Konsistenz wiederhergestellt werden.

NOPASSWD zeigt Inkonsistenz der Datei /etc/passwd an.

18.11.1.2 TSTAENT(S)

Name:

tstaent - test and enter control file

Definition

```
#include <dasi.h>

tstaent( ctl_file )
char *ctl_file;
```

Beschreibung

Voraussetzung ist die Existenz der Steuerdatei ctl_file, welche mit "smutil -c" erstellt und initialisiert wurde.

Mit der Funktion "tstaent" hinterlegt der aufrufende Benutzer in der Steuerdatei eines zentralen Datenbestandes oder in der globalen Steuerdatei eines verteilten Datenbestands einen Benutzereintrag. Letzterer besteht aus der User-ID des Benutzers, dem Login-Namen des ersten Beschreibungssatzes zur User-ID in der Datei /etc/passwd und dem Stationsnamen. Der Eintrag wird verwehrt, wenn durch "testalock" bzw. "sminit" zur Datensicherung aufgefordert wurde.

Rückgabewert

Bei gültig erfolgtem Benutzereintrag liefert "tstaent" den Wert 0 zurück.

Bei Fehlern, welche das Betriebssystem entdeckt, wird -1 zurückgegeben und die Variable errno gesetzt.

In folgenden Fällen bricht die Funktion ab und gibt eine der in /usr/include/dasi.h definierten Statusmeldungen zurück:

NOACCESS	Kein exklusiver Zugriff auf die Steuerdatei möglich.
ILLCTLFILE	Ungültige Steuerdatei.

NOUSERMOD Der Datenbestand ist gegenwärtig zur Bearbeitung gesperrt.

TOOMANY Ein weiterer Benutzereintrag ist zur Zeit nicht zulässig.

NOPASSWD kein Eintrag in /etc/passwd gefunden.

Dateien

/usr/include/dasi.h enthält die Statusmeldungen
/etc/passwd enthält Login-Namen

siehe auch

leavesys(S), testalock(C), resetlock(C), sminit(C), smexit(C), smutil(C).

Bemerkung

Die Statusmeldung NOPASSWD zeigt Inkonsistenz der Datei /etc/passwd an.

18.11.2 Kommandos

18.11.2.1 CREADAS(C)

Name:

creadas - legt menügesteuert ein GLOBAL-Control-File und alle dazugehörigen LOCAL-Control-Files an.

Aufruf

creadas

Beschreibung

Nach dem Aufruf erscheint am Bildschirm:

Anlegen des GLOBAL-Control-Files und aller LOCAL-Control-Files

Pfadname des GLOBAL-Control-Files :

Hier ist der vollständige Pfadname anzugeben z.B.:
/../ST008/usr/adm/DSkomm

DS_NAME des GLOBAL-Control-Files:

Der DS_Name ist frei wählbar. Sinnvoll ist es, einen Bezug zu Pfadnamen des GLOBAL-Control-Files herzustellen.
z.B.: DSKOMM

maximale Anzahl der Benutzer:

Hier gibt man, an wieviele Benutzer sich maximal im GLOBAL-Control-File eintragen dürfen.

Sind alle Angaben richtig? (j/n):

Werden die Angaben bestätigt, wird das GLOBAL-Control-File angelegt oder eine entsprechende Fehlermeldung ausgegeben.

Das GLOBAL-Control-File wude fehlerfrei angelegt.

Nach erfolgreichem Anlegen des GLOBAL-Control-Files, können die Pfadnamen für alle LOCAL-Control-Files einge-

geben werden. Danach werden alle LOCAL-Control-Files angelegt und eine Erfolgs- oder Fehlermeldung ausgegeben.

Hinweis: creadas legt das GLOBAL-Control-File mit der Versionsnummer 1 an. Soll nachträglich im GLOBAL-Control-Files etwas geändert werden, muß das Utility "smutil" (siehe DASI-Handbuch) benützt werden.
Es gelten die Einschränkungen der UNIX Screen-IO.

Beispiel für eine Maskendatei

```
04,58  Anzahl Kommandos, Länge Tabellenglied
sanmeld; echo -n "\n\033N5Auslösetaste\03305 "; read ein
logsich; echo -n "\n\033N5Auslösetaste\03305 "; read ein
physich; echo -n "\n\033N5Auslösetaste\03305 "; read ein
sabmeld; echo -n "\n\033N5Auslösetaste\03305 "; read ein
20,33
```

Wählen Sie die Art der Datensicherung
==========

1 = Datensicherung anmelden
2 = Logische Datensicherung
3 = Physische Datensicherung
4 = Datensicherung abmelden
0 = Ende (FE, ABBR)

Auswahl:

18.11.2.2 DAS(C)

Name:

das - Menüprogramm für Datensicherung

Aufruf

das mask_file

Parameter

mask_file Name der Steuerdatei

Beschreibung

"das" entnimmt der Steuerdatei Informationen über Anzahl und Länge der Kommandozeilen, die Kommandozeilen selbst, das gewünschte Menübild sowie die Eingabeposition (Cursorzeile, Cursorspalte), zeigt das Menü an, nimmt vom Benutzer über zugeordnete Nummern Kommandos entgegen und führt diese aus.

Syntax der Steuerdatei

Zeile 1		: ⟨nn⟩
	⟨nn⟩	: Anzahl Kommandozeilen (2 Stellen)
Zeile 2 bis ⟨nn⟩ + 1		: Kommandozeilen. Diese Zeilen dürfen eine Länge von maximal 79 Zeichen haben.
Zeile ⟨nn⟩ + 2		: ⟨zz⟩,⟨ss⟩
	⟨zz⟩	: Zeile für die Cursorposition (2 Stellen)
	⟨ss⟩	: Spalte für die Cursorposition (2 Stellen)
Anschließende Zeilen		: Menübild (erscheint am Bildschirm)

Beispiel für eine Steuerdatei

```
03 Anzahl Kommandos
KOMMERZ_A; echo -n "\n\033N5Auslösetaste\03305 "; read a
KOMMERZ_B; echo -n "\n\033N5Auslösetaste\03305 "; read a
SCHRBTSCH; echo -n "\n\033N5Auslösetaste\03305 "; read a
24,27
```

D A T E N S I C H E R U N G

Wählen Sie unter folgenden Kommandos

1 = Datensicherung Kommerzpaket A

2 = Datensicherung Kommerzpaket B

3 = Datensicherung Elektron Schreibtisch

0 = Ende (FE, ABBR)

Auswahl:

18.11.2.3 RESETLOCK(C)

Name:

resetlock - Rücksetzen Sperre von Datenbasis in Steuerdatei

Aufruf

```
resetlock [-lv] ctl_file
```

Optionen

-l liefert die für das Sichern beschriebene Funktionalität auch für das Laden eines gesicherten Datenbestands (load) - wird in der zweiten Ausbaustufe der Datensicherungswerkzeuge realisiert.

-v Meldungen werden zusätzlich auf stdout ausgegeben (verbose).

Parameter

ctl_file Name der Steuerdatei

Beschreibung

"resetlock" hebt die in der Steuerdatei gesetzte Sperre für einen zu sichernden Datenbestand wieder auf.

Bei zentralem Datenbestand setzt "resetlock" in der Steuerdatei den Sicherungszustand "SAVMOD" um auf "USMOD" (vergleiche testalock(C)). Bei verteiltem Datenbestand schaltet "resetlock" in den lokalen Steuerdateien die Sicherungsphase "SAVMOD" auf SL_END" (siehe sminit(C)). In beiden Fällen erhöht "resetlock" die Versionsnummer und zeigt sie an.

Die aktuelle Versionsnummer wird ausgegeben, wenn "resetlock" in einem anderen Sicherungszustand als "SAVMOD" aufgerufen wird.

Die globale Steuerdatei des verteilten Datenbestands darf nicht mit "resetlock" bearbeitet werden.

Alle Meldungen werden auf stderr ausgegeben.

Exit Status

Bei erfolgreich aufgehobener Sperre gibt "resetlock" den exit status 0 zurück. Anderenfalls wird ein von 0 verschiedener Wert geliefert:

1 no access (kein exklusiver Zugriff auf die Steuerdatei möglich)

145 illegal type (falscher Typ der Steuerdatei)

146 illegal mode (falsche Verarbeitungsart der Steuerdatei)

Weitere von 0 verschiedene Werte zeigen Ausnahmefälle wie etwa falsche Dateigröße oder Systemfehler wie z.B. open error usw. an.

Siehe auch

tstaent(S), leavesys(S), testalock(C), sminit(C), smexit(C), smutil(C).

18.11.2.4 SMEXIT(C)

Name:

smexit - save manager: Datensicherung abschließen

Aufruf

```
smexit [-lv] ctl_file list_loc_ctlf
```

Optionen

-l liefert die für das Sichern beschriebene Funktionalität auch für das Laden eines gesicherten Datenbestands (load) - wird in der zweiten Ausbaustufe der Datensicherungswerkzeuge realisiert.

-v Meldungen werden zusätzlich auf stdout ausgegeben (verbose).

Parameter

ctl_file	Name der Steuerdatei
list_loc_ctlf	Liste der Namen aller lokalen Steuerdateien

Beschreibung

Mit "smexit" setzt der save manager die Steuerdateien eines verteilten Datenbestands nach beendeter Sicherung zurück, so daß der Datenbestand von den Benutzern wieder bearbeitet werden kann.

In den lokalen Steuerdateien schaltet "smexit" die Sicherungsphase "SL_END" auf "USMOD" (siehe sminit(C)). Sind alle lokalen Steuerdateien im Endzustand, setzt "smexit" in der globalen Steuerdatei den Sicherungsgszustand "SAVMOD" um auf "USMOD", erhöht die Versionsnummer und zeigt sie an.

Die aktuelle Versionsnummer wird ausgegeben, wenn "smexit" in einem anderen Sicherungszustand als "SAVMOD" aufgerufen wird.

Alle Meldungen werden auf stderr ausgegeben.

Exit Status

Bei erfolgreichem Abschluß gibt "smexit" den exit status 0 zurück. Anderenfalls wird ein von 0 verschiedener Wert geliefert:

1 no access (kein exklusiver Zugriff auf die Steuerdatei möglich)

141 mehrere vorher gemeldete Fehler oder Inkonsistenzen bei der Abarbeitung der Liste aller lokalen Steuerdateien

145 illegal type (falscher Typ der Steuerdatei)

146 illegal mode (falsche Verarbeitungsart der Steuerdatei)

Weitere von 0 verschiedene Werte zeigen Ausnahmefälle wie etwa falsche Dateigröße oder Systemfehler wie z.B. open error usw. an.

Siehe auch

tstaent(S), leavesys(S), testalock(C), resetlock(C), sminit(C), smutil(C).

18.11.2.5 SMINIT(C)

Name:

sminit - save manager: Initialisiere Datensicherung

Aufruf

```
sminit [-iltv] ctl_file list_loc_ctlf
```

Optionen

-i Falls Datensicherung nicht möglich ist, werden die angemeldeten Benutzer mit dem Login-Namen und der Station ausgegeben (info).

-l liefert die für das Sichern beschriebene Funktionalität auch für das Laden eines gesicherten Datenbestands (load) - wird in der zweiten Ausbaustufe der Datensicherungswerkzeuge realisiert.

-t Nur in Verbindung mit Option -i: Zum Login-Namen und zur Station wird zusätzlich der tty-Name ermittelt und ausgegeben (tty name).

-v Meldungen werden zusätzlich auf stdout ausgegeben (verbose).

Parameter

ctl_file Name der Steuerdatei

list_loc_ctlf Liste der Namen aller lokalen Steuerdateien

Beschreibung

Mit "sminit" initialisiert der save manager die Steuerdateien eines verteilten Datenbestandes, so daß dieser gesichert werden kann.

Bei verteiltem Datenbestand (mit seinen lokalen Steuerdateien und seiner globalen Steuerdatei) koordiniert die Kommandofolge "sminit -- testalock -- resetlock -- smexit" in den lokalen Steuerdatein den Sicherungszyklus "SAVREQ -- SAVMOD -- SL_END -- USMOD". Gleich-

zeitig dokumentieren "sminit" und "smexit" in der globalen Steuerdatei die Startphase "SAVREQ" bzw. "SAVMOD" und die Abschlußphase "USMOD" der Datensicherung.

Bei zentralem Datenbestand kann die globale und die (dann einzige) lokale Steuerdatei zu einer Steuerdatei zusammengefaßt werden, für die dann "testalock" (siehe dort) und "resetlock" die Aufgaben von "sminit" und "smexit" mitübernehmen.

Mit der Option -i zeigt "sminit" wie "smutil -i" etwaige aktive Benutzer an.

Alle Meldungen werden auf stderr ausgegeben.

Exit Status

Bei erfolgreicher Initialisierung gibt "sminit" den exit status 0 zurück. Anderenfalls wird ein von 0 verschiedener Wert geliefert:

1 no access (kein exklusiver Zugriff auf die Steuerdatei möglich)

121 active users (es sind noch aktive Benutzer am zu sichernden Datenbestand)

141 mehrere vorher gemeldete Fehler oder Inkonsistenzen bei der Abarbeitung der Liste aller lokalen Steuerdateien

145 illegal type (falscher Typ der Steuerdatei)

146 illegal mode (falsche Verarbeitungsart der Steuerdatei)

Weitere von 0 verschiedene Werte zeigen Ausnahmefälle wie etwa falsche Dateigröße oder Systemfehler wie z.B. open error usw. an.

Dateien

/etc/utmp enthält den tty-Namen

Siehe auch

tstaent(S), leavesys(S), testalock(C), resetlock(C), smexit(C), smutil(C).

18.11.2.6 SMUTIL(C)

Name:

smutil - save manager: Utility zur Verwaltung der Datensicherungs-Steuerdateien

Aufruf

```
smutil -c ctl_file ds_name max_user ds_type vers_no
smutil -d ctl_file login_name station_id
smutil -i [-t] ctl_file
smutil -p [-f ds_type] [-m ds_mode] [-n ds_name] [-u max_user]
                                    [-v vers_no] ctl_file
smutil -s ctl_file list_loc_ctlf
```

Optionen

- -c Steuerdatei anlegen (create)
- -d Benutzereintrag löschen (delete)
- -i Inhalt der Steuerdatei ausgeben (info)
- -t Nur in Verbindung mit Option -i: Die tty-Namen eingetragener Benutzer ermitteln und zusätzlich mit ausgeben (tty name)
- -p Verwaltungseinträge in der Steuerdatei ändern (patch)
- -f Nur in Verbindung mit Option -p: Den Typ der Steuerdatei ändern (file type)
- -m Nur in Verbindung mit Option -p: Die Verarbeitungsart der Steuerdatei ändern (mode)
- -n Nur in Verbindung mit Option -p: Den Namen des Datenbestands ändern (name)
- -u Nur in Verbindung mit Option -p: Die maximal zulässige Benutzeranzahl ändern (user maximum)
- -v Nur in Verbindung mit Option -p: Die Versionsnummer ändern (version)

-s Setze lokale Steuerdateien (set local)

Parameter

ctl_file	Name der Steuerdatei		
list_loc_ctlf	Liste der Namen aller lokalen Steuerdateien		
login_name	Namenskomponente im Benutzereintrag		
station_id	Stationskomponente im Benutzereintrag		
ds_name	Name des Datenbestands		
ds_type	Typ der Steuerdatei		
	GLOBAL	0	globale Steuerdatei
	LOCAL	1	lokale Steuerdatei
vers_no	Versionsnummer		
ds_mode	Bearbeitungsart des Datenbestands		
	USMOD	0	Bearbeitung zulässig
	SAVREQ	1	Sicherung angemeldet
	SAVMOD	2	Sicherung zulässig
	LOADREQ	3	Laden angemeldet
	LOADMOD	4	Laden zulässig
	SL_END	5	Sicherung/Laden beendet
max_user	maximale Anzahl der Benutzereinträge		

Beschreibung

Mit "smutil" erstellt und verwaltet der save manager die Steuerdateien.

smutil -c:

Dieses Kommando erstellt die Steuerdatei für einen zentralen Datenbestand oder die globale Steuerdatei für einen verteilten Datenbestand. Beide müssen mit dem Typ GLOBAL angelegt werden. Die lokalen Steuerdateien eines verteilten Datenbestands werden mit "smutil -s"

eingerichtet.

smutil -d:

Hiermit kann die Liste der Benutzereinträge korrigiert werden. Fehlt die Stationsangabe im Benutzereintrag, wie z.B. bei nicht vernetzem System, muß als Stationsname der Leerstring "" eingegeben werden.

smutil -i:

Mit dieser Auskunftsfunktion können alle Verwaltungseinträge und ggf. Benutzereinträge einer Steuerdatei aufgelistet werden. Bei mehreren Beschreibungssätzen in der Datei /etc/passwd zur selben User_Id muß der Benutzer mit dem Login-Namen des ersten Beschreibungssatzes eingeloggt sein, wenn der Name des Login-Terminals angezeigt werden soll.

smutil -p:

Diese Funktion dient der Korrektur von Verwaltungseinträgen. Magic number und Anzahl aktueller Benutzer sind davon ausgeschlossen.

smutil -s:

Mit diesem Kommando werden die in der Liste genannten lokalen Steuerdateien erstellt und initialisiert. Die globale Steuerdatei muß vorhanden sein. Globale Steuerdateien einer beliebigen gültigen Sicherungsphase werden akzeptiert. Bereits existierende lokale Steuerdateien werden nicht verändert. So ist ein Wiederaufsetzen des Sicherungslaufs bei lokalen Ausnahmesituationen in jeder Phase möglich.

Exit Status

Bei erfolgreicher Bearbeitung gibt "smutil" den exit status 0 zurück.

Anderenfalls werden als exit status dieselben von 0 verschiedenen Werte wie bei den Kommandos testalock, resetlock, sminit und smexit geliefert und eine Meldung auf stderr ausgegeben.

Dateien

/etc/utmp enthält den tty-Namen

Siehe auch
tstaent(S), leavesys(S), testalock(C), resetlock(C), sminit(C), smexit(C)

18.11.2.7 TESTALOCK(C)

Name:

testalock - Teste und sperre Datenbasis in Steuerdatei

Aufruf

testalock [-iltv] ctl_file

Optionen

-i Falls Datensicherung nicht möglich ist, werden die angemeldeten Benutzer mit dem Login-Namen und der Station ausgegeben (info).

-l liefert die für das Sichern beschriebene Funktionalität auch für das Laden eines gesicherten Datenbestands (load) - wird in der zweiten Ausbaustufe der Datensicherungswerkzeuge realisiert.

-t Nur in Verbindung mit Option -i: Zum Login-Namen und zur Station wird zusätzlich der tty-Name ermittelt und ausgegeben (tty name)

-v Meldungen werden zusätzlich auf stdout ausgegeben (verbose).

Parameter

ctl_file Name der Steuerdatei

Beschreibung

"testalock" testet einen zu sichernden Datenbestand auf exklusiven Zugriff und setzt eine Sperre in der Steuerdatei.

Bei zentralem Datenbestand koordiniert die Kommandofolge "testalock -- resetlock" in der (einzigen) Steuerdatei den Sicherungszyklus "SAVREQ -- SAVMOD -- USMOD". In diesem Falle kann "testalock" mit Option -i die Funktion von "smutil -i" mit ausführen, nämlich etwaige aktive Benutzer anzuzeigen.

Bei verteiltem Datenbestand schaltet "testalock" im Rahmen der Kommandofolge "sminit -- testalock -- resetlock -- smexit" (siehe sminit(C)) in den lokalen Steuerdateien die Sicherungsphase "SAVREQ" auf "SAVMOD".

Die globale Steuerdatei des verteilten Datenbestands darf nicht mit "testalock" bearbeitet werden.

Alle Meldungen werden auf stderr ausgegeben

Exit Status

Bei erfolgreich zurückgesetzter Sperre gibt "testalock" den exit status 0 zurück. Anderenfalls wird ein von 0 verschiedener Wert geliefert.

1 no access (kein exklusiver Zugriff auf die Steuerdatei möglich)

121 active users (es sind noch aktive Benutzer am zu sichernden Datenbestand)

145 illegal type (falscher Typ der Steuerdatei)

146 illegal mode (falsche Verarbeitungsart der Steuerdatei)

Weitere von 0 verschiedene Werte zeigen Ausnahmefälle wie etwa falsche Dateigröße oder Systemfehler wie z.B. open error usw. an.

Siehe auch

tstaent(S), leavesys(S), resetlock(C), sminit(C), smexit(C), smutil(C).

18.11.3 DASI-Fallbeispiel

Voraussetzungen

Konfiguration

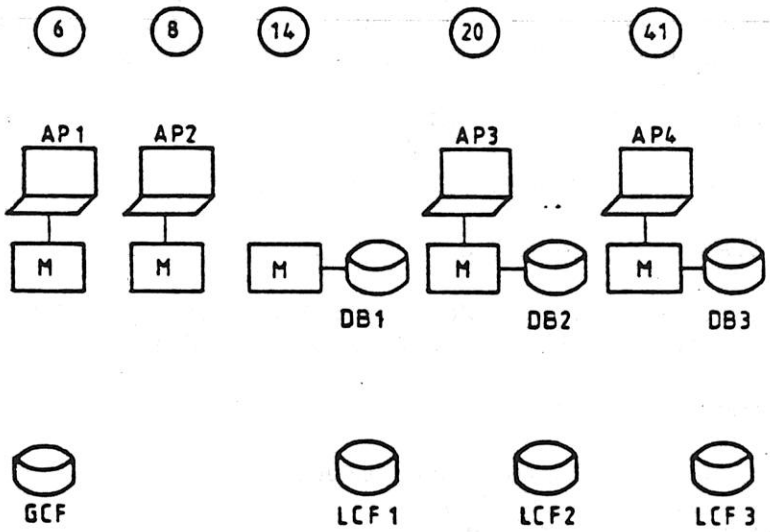

Abbildung 18-10: Beispielkonfiguration

Beschreibung

Die Datenbestände liegen im Netz verteilt auf den Stationen 14, 20 und 41. Anwendungen werden von beliebigen Arbeitsplätzen aus durchgeführt.

Anmerkung 1
Bei verteilten Daten liegt ein Teildatenbestand und die zugeordnete lokale Steuerdatei stets an demselben Netzknoten. Die globale Steuerdatei kann an einem beliebigen Netzknoten installiert werden. Ein zentraler Datenbestand und die zugeordnete Steuerdatei liegen auf demselben Netzknoten.

Anmerkung 2
Es empfiehlt sich, die globale Steuerdatei im "root" file system eines Netzknotens anzulegen, damit dieses file system nicht unbeabsichtigt abgemeldet werden kann. Der Name einer Steuerdatei wird sinnvollerweise in das Directory "/usr/adm" eingetragen.

Anlegen der Steuerdateien

Beispielsweise sollen von Station 20 aus die globale Steuerdatei auf Station 6, die lokalen Steuerdateien auf den Stationen 14, 20 und 41 angelegt werden. Die Stationsnamen seien etwa ST006, ST008 usw.

globale Steuerdatei anlegen:

```
# smutil -c /../ST006/usr/adm/DBtest DB_TEST 12 GLOBAL 1001
```

lokale Steuerdateien setzen:

```
# smutil -s /../ST006/usr/adm/DBtest /../ST014/usr/adm/DBtest14
    /usr/adm/DBtest20 /../ST041/usr/adm/DBtest41
```

Ergebnisse anzeigen:

```
# smutil -i /../ST006/usr/adm/DBtest
data set name:      DB_TEST
data set type:      GLOBAL
data set mode:      USMOD
version number:     1001
maximal users:      12
actual users:       0

# smutil -i /../ST014/usr/adm/DBtest14
data set name:      DB_TEST
data set type:      LOCAL
data set mode:      USMOD
version number:     1001
maximal users:      12
actual users:       0
```

usw.

Beispiel für den typischen Organisationsablauf

Anmeldungen aus den Applikationsprogrammen

Eine Applikation wird etwa an der Station 8 durchgeführt:

```
main()
{         /* Applikation 1 */
      .
      .
      tstaent( "/../ST006/usr/adm/DBtest" ) ; /* anmelden */
      .
      .
      .
}
```

Eine weitere Applikation wird z.B. von Station 41 gestartet:

```
main()
{         /* Applikation 2 */
      .
      .
      .
      tstaent( "/../ST006/usr/adm/DBtest" ) ; /* anmelden */
      .
      .
      .
}
```

Auskunft über den aktuellen Stand gibt an Station 6:

```
# smutil -it /usr/adm/DBtest
data set name:       DB_TEST
data set type:       GLOBAL
data set mode:       USMOD
version number:      1001
maximal users:       12
actual users:        2
LOGIN NAME     STATION        TTY NAME
root           ST008          console
root           ST041          console
```

Beachte!

Die Datei /etc/passwd kann mehrere Sätze enthalten, welche dieselbe "User-Id" beschreiben. Ein Benutzer wird mit dem Login-Namen seines ersten Beschreibungssatzes in die Steuerdatei eingetragen. Er muß mit diesem Namen eingeloggt sein, wenn mit Hilfe eines DASI-Kommandos sein Login-Terminal ermittelt werden soll.

Hinweis

Ein Programmbeispiel für Testzwecke ist dem gleichlautenden Kapitel dieses Fallbeispiels zu entnehmen.

Abmeldungen aus den Applikationsprogrammen

Applikation 1 ist gelaufen und meldet sich ab:

```
main()
{  .       /* Applikation 1 */
      .
      .
      .
      tstaent( "/../ST006/usr/adm/DBtest" );
      .
      .

      .    .
      leavesys( "/../ST006/usr/adm/DBtest" ); /* abmelden */
      .
      .
}
```

Auskunft an Station 6

```
# smutil -it /usr/adm/DBtest
data set name:       DB_TEST
data set type:       GLOBAL
data set mode:       USMOD
version number:      1001
maximal users:       12
actual users:        1

LOGIN NAME      STATION         TTY NAME
root            ST041           console
```

Applikation 2 ist gelaufen und meldet sich ab:

```
main()
{          /* Applikation 2 */
      .
      .
      .
      tstaent( "/../ST006/usr/adm/DBtest" );
      .
      .

      .
      leavesys( "/../ST006/usr/adm/DBtest" ); /* abmelden */
      .
      .
}
```

Auskunft an Station 6

```
# smutil -it /usr/adm/DBtest
data set name:      DB_TEST
data set type:      GLOBAL
data set mode:      USMOD
version number:     1001
maximal users:      12
actual users:       0
```

Datensicherung initialisieren

Die Datensicherung wird nun z.B. von Station 6 aus initialisiert und überwacht:

```
# sminit -it /usr/adm/DBtest /../ST014/usr/adm/DBtest14
/../ST020/usr/adm/DBtest20 /../ST041/usr/adm/DBtest41
```

Ergebnis:

```
# smutil -it /usr/adm/DBtest
data set name:      DB_TEST
data set type:      GLOBAL
data set mode:      SAVMOD
version number:     1001
maximal users:      12
actual users:       0


# smutil -i /../ST014/usr/adm/DBtest14
data set name:      DB_TEST
data set type:      LOCAL
data set mode:      SAVREQ
version number:     1001
maximal users:      12
actual users:       0
```

usw.

Lokales Sichern

Alle Teildatenbestände werden nun jeweils an ihren Stationen gesichert. Für Station 20 laufe etwa folende skizzierte Sicherungsprozedur:

```
.
.
testalock /usr/adm/DBtest20
.
.

.
.
resetlock /usr/adm/DBtest20
.
.
```

Im Abschnitt Musterprozeduren 18.11.5 sind ausführliche Beispiele angegeben. Ersetze dort die allgemeine Bezeichnung LOCAL-CONTROLFILE durch /usr/adm/DBtest20.

Entsprechende Prozeduren seien für Stationen 14 und 41 aktiv.

Über den Stand der Datensicherung können nun z.B. an Station 6 folgende Informationen ausgegeben werden:

```
# smutil -i /../ST014/usr/adm/DBtest14
data set name:      DB_TEST
data set type:      LOCAL
data set mode:      SAVMOD
version number:     1001
maximal users:      12
actual users:       0
```

Station 14 hat mit der Sicherung begonnen, testalock hat den Request Mode in den Save Mode umgeschaltet.

```
# smutil -i /../ST020/usr/adm/DBtest20
data set name:      DB_TEST
data set type:      LOCAL
data set mode:      SL_END
version number:     1002
maximal users:      12
actual users:       0
```

Station 20 ist mit der Sicherung fertig, resetlock hat auf Save End geschaltet und die Versionsnummer erhöht.

```
# smutil -i /../ST041/usr/adm/DBtest41
data set name:      DB_TEST
data set type:      LOCAL
data set mode:      SAVREQ
version number:     1001
maximal users:      12
actual users:       0
```

Station 41 hat mit der Sicherung noch nicht angefangen.

Datensicherung abschließen

Nachdem alle lokalen Sicherungen abgeschlossen sind, gebe Station 6 den verteilten Datenbestand zur Bearbeitung frei:

```
# smexit /usr/adm/DBtest /../ST014/usr/adm/DBtest14
```

/../ST020/usr/adm/DBtest20 /../ST041/usr/adm/DBtest41

Die Steuerdateien sind dann mit erhöhter Versionsnummer wieder im User Mode:

```
# smutil -it /usr/adm/DBtest
data set name:      DB_TEST
data set type:      GLOBAL
data set mode:      USMOD
version number:     1002
maximal users:      12
actual users:       0

# smutil -i /../ST014/usr/adm/DBtest14
data set name:      DB_TEST
data set type:      LOCAL
data set mode:      USMOD
version number:     1002
maximal users:      12
actual users:       0
```

usw.

18.11.4 Programmbeispiel für Testzwecke

Für Testzwecke können aus den Bibliotheksfunktionen wie folgt ablauffähige Kommandos gebildet werden:

```
#include <dasi.h>

main(argc, argv)
int argc;
char **argv;
{
     switch ( tstaent( argv[ 1 ] ) )
     {
     case 0:
          exit( 0 );
     case NOACCESS:
          printf( "exklusiver Zugriff nicht möglich\n" );
          exit( NOACCESS );
     case ILLCTLFILE:
          printf( "ungültige Steuerdatei\n" );
          exit( ILLCTLFILE );
     case NOUSERMOD:
          printf( "Dateibestand zur Bearbeitung gesperrt\n" );
          exit( NOUSERMOD );
     case TOOMANY:
          printf( "zu viele Benutzereinträge\n" );
          exit( TOOMANY );
     case NOPASSWD:
          printf( "Eintrag in /etc/passwd nicht gefunden\n" );
          exit( NOPASSWD );
     case -1:
          printf( "system error errno %d\n" , errno );
          exit( -1 );
     default:
          printf( "unknown error\n" );
          exit( -1 );
     }
}
```

18.11.5 Musterprozeduren

Die notwendigen Schritte können anhand der beiden folgenden Shell-Scripts nachvollzogen werden. Selbstverständlich handelt es sich dabei um reine Musterbeispiele, die den speziellen Anwendungen angepaßt werden müssen.

18.11.5.1 Beispiel einer physikalischen Datensicherung

```
################################################
# prüfen, ob Sicherung erfolgen soll und kann
################################################

     testalock LOCAL-CONTROLFILE 2)/dev/null
     a=$?
     # Return-Code
     case $a in
       0) ;;
       1) echo "auf die Steuerdatei wird gerade zugegriffen"
          echo "bitte Aufruf wiederholen"
          exit;;
     121) echo "es sind noch Benutzer aktiv"
          exit;;
     146) echo "es ist keine Sicherung angeordnet"
          exit;;
       *) echo "Steuerdatei inkonsistent"
          echo "Systemverwalter ansprechen"
          exit;;
     esac


################
# Netz abmelden
################

     # Netz aktiv ?

     a=`/etc/netstat 2)/dev/null`
     if [ "$a" ]
     then
        echo "Netz noch aktiv"
        exit
     fi
     # Netz abmelden
```

```
      /bin/stopnet 2>/dev/null
      a=$?
      if [ $a -ne 0 ]
      then
        echo "stopnet nicht durchführbar"
        exit
      fi


#########################
# Filesystem abmelden
#########################

      /etc/umount /dev/di1data1>/dev/null
      a=$?
      if [ $a -ne 0 ]
      then
         echo "umount nicht durchführbar"
         /bin/startnet 1>/dev/null 2>/dev/null
         a=$?
         if [ $a -ne 0 ]
         then
            echo "startnet nicht durchführbar"
         fi
         exit
      fi


#########################
# Datenträger anmelden
#########################

      stctrl -l /dev/rst0 1>/dev/null
      a=$?
      if [ $a -ne 0 ]
      then
         echo "Streamer nicht ladbar !"
         /etc/mount  /dev/di1data1 /data1 2>/dev/null
         a=$?
         if [ $a -ne 0 ]
         then
            echo "mount nicht durchführbar"
         fi
         /bin/startnet 1>/dev/null 2>/dev/null
         a=$?
```

```
        if [ $a -ne 0 ]
        then
           echo "startnet nicht durchführbar"
        fi
        exit
     fi


#################################
# Physikalische Datensicherung
#################################

     echo "\n+++++ DASI begonnen: " >>/usr/adm/PROTOKOLL
     date >>/usr/adm/PROTOKOLL
     /etc/dicopy /dev/rdi1data1 /dev/rst0 1>/tmp/DASIFEH
     a=$?
     if [ $a -ne 0 ]
     then
        cat /tmp/DASIFEH >>/usr/adm/PROTOKOLL
        echo "!!!!! DASI fehlerhaft beendet !!!!!\n" >>/usr/adm/PROTOKOLL
        echo "DASI fehlerhaft !"
        cat /tmp/DASIFEH
     else
        # Sicherungslauf ok
        # Prüflesen (VERIFY)
        /etc/dicopy -c /dev/rdi1data1 /dev/rst0 1>/tmp/DASIFEH
        a=$?
        if [ $a -ne 0 ]
        then
           cat /tmp/DASIFEH >>/usr/adm/PROTOKOLL
           echo "!!!!! DASI VERIFY fehlerhaft beendet !!!!!\n" >>/usr/adm/PROTOK
           echo "DASI VERIFY fehlerhaft !"
           cat /tmp/DASIFEH
        else
        #  Sicherungslauf ok; Prüflesen ok
```

```
#############################################################################
#  Rücksetzen der Sicherungsanforderung in der lokalen Steuerdatei
#############################################################################

          resetlock  LOCAL-CONTROLFILE 2>/dev/null
          a=$?
          if [ $a -ne 0 ]
          then
          #   1  kein exclusiver Zufgriff auf die Steuerdatei
          # 146  keine Sicherung angeordnet
             echo "Steuerdatei INKONSISTENT"
             echo "Fehler: RESETLOCK-exit=$a\n" >>/usr/adm/PROTOKOLL
          else
             echo "DASI korrekt beendet"
          #  Versionsnummer (von resetlock)
          set `resetlock -v LOCAL-CONTROLFILE 2>/dev/null`
          eval VERSNR=$"$#"
          echo "\033N5Versionsnummer $VERSNR aufkleben !!\03305"
          echo "Versionsnummer $VERSNR:\n" >>/usr/adm/PROTOKOLL
          echo "++++ Datenträger erfolgreich erstellt\n" >>/usr/adm/PROTOKOLL
          fi
       fi
    fi


########################
# Datenträger entladen
########################

    stctrl -u /dev/rst0 1>/dev/null


######################
# Filesystem mounten
######################

    /etc/mount /dev/di1data1 /data1 2>/dev/null
    a=$?
    if [ $a -ne 0 ]
    then
       echo "mount nicht durchführbar"
    fi
```

```
################
# Netz starten
################

     /bin/startnet 1>/dev/null 2>/dev/null
     a=$?
     if [ $a -ne 0 ]
     then
        echo "startnet nicht durchführbar"
     fi
```

18.11.5.2 Beispiel einer logischen Datensicherung

```
#################################################
# prüfen, ob Sicherung erfolgen soll und kann
#################################################

     testalock LOCAL-CONTROLFILE 2>/dev/null
     a=$?
     # Return-Code
     case $a in
       0) ;;
       1) echo "auf die Steuerdatei wird gerade zugegriffen"
          echo "bitte Aufruf wiederholen"
          exit;;
     121) echo "es sind noch Benutzer aktiv"
          exit;

     146) echo "es ist keine Sicherung angeordnet"
          exit;;
       *) echo "Steuerdatei inkonsistent"
          echo "Systemverwalter ansprechen"
          exit;;
     esac
```

```
########################
# Datenträger anmelden
########################

     stctrl -l /dev/rst0 1>/dev/null
     a=$?
     if [ $a -ne 0 ]
     then
        echo "Streamer nicht ladbar !"
        exit
     fi


##################
# Datensicherung
##################

     echo "\n+++++ DASI begonnen: " >>/usr/adm/PROTOKOLL
     date >>/usr/adm/PROTOKOLL
     lback -b FILE1 FILE2 /dev/rst0 1>/dev/null 2>/tmp/DASIFEH
     a=$?
     if [ $a -ne 0 ]
     then
        cat /tmp/DASIFEH >>/usr/adm/PROTOKOLL
        echo "!!!!! DASI fehlerhaft beendet !!!!!\n" >>/usr/adm/PROTOKOLL
        echo "DASI fehlerhaft !"
        cat /tmp/DASIFEH
     else

#############################################################################
#  Rücksetzen der Sicherungsanforderung in der lokalen Steuerdatei
#############################################################################

        resetlock  LOCAL-CONTROLFILE 2>/dev/null
        a=$?        if [ $a -ne 0 ]
        then
        #   1  kein exclusiver Zugriff auf die Steuerdatei
        # 146  keine Sicherung angeordnet
           echo "Steuerdatei INKONSISTENT"
           echo "Fehler: RESETLOCK-exit=$a\n" >>/usr/adm/PROTOKOLL
        else
           echo "DASI korrekt beendet"
           #  Versionsnummer (von resetlock)
           set `resetlock -v LOCAL-CONTROLFILE 2>/dev/null`
```

```
            eval VERSNR=$"$#"
            echo "\033N5Versionsnummer $VERSNR aufkleben !!\03305"
            echo "Versionsnummer $VERSNR:" >>/usr/adm/PROTOKOLL
            echo "++++ Datenträger erfolgreich erstellt\n" >>/usr/adm/PROTOKOLL
         fi
      fi


#########################
# Datenträger entladen
#########################

      stctrl -u /dev/rst0 1>/dev/null
```

18.11.6 Muster für Sicherung von Kommerz-Dateien

```
##################################
#  Anmelden der Datensicherung
##################################

     G=/usr/adm/DSkomm                                          ########
     L8=/usr/adm/DSkomm08                                       ########
     L25=/../ST025/usr/adm/DSkomm25                             ########
     sminit $G $L8 $L25 2>/dev/null
     a=$?
       case $a in
          0) echo Die Datensicherung wurde angemeldet
             exit;;
          1) echo "Kein Zugriff auf das Control-File möglich"
             exit;;
        121) echo Es sind noch User aktiv
             sminit -i $G $L8 $L25
             exit;;
        141) echo Fehler bei der Abarbeitung aller LOCAL-Control-Files
             exit;;
        145) echo Falsches Control-File
             exit;;
        146) echo Falsche Verarbeitungsart des Control-Files
             exit;;
          *) echo system error
             exit;;
       esac


################################
# Abmelden der Datensicherung
################################

     G=/usr/adm/DSkomm                                          ########
     L8=/usr/adm/DSkomm08                                       ########
     L25=/../ST025/usr/adm/DSkomm25                             ########
     smexit $G $L8 $L25 2>/dev/null
     a=$?
        case $a in
          0) echo Die Datensicherung wurde abgemeldet
             exit;;
          1) echo Kein Zugriff auf das Control-File möglich
             exit;;
```

```
        121) echo Es sind noch User aktiv
             exit;;
        141) echo Datensicherung noch nicht beendet
             smexit $G $L8 $L25
             exit;;
        145) echo Falsches Control-File
             exit;;
        146) echo Falsche Verarbeitungsart des Control-Files
             exit;;
          *) echo system error
             exit;;
        esac
```

18.11.6.1 Beispiel eines Shell-Script für logische Datensicherung

```
#########################################################
# prüfen ob die Sicherung durchgeführt werden kann
#########################################################

    testalock /usr/adm/DSkomm08 2>/dev/null                    ########
    a=$?
       case $a in
        0) ;;
        1) echo auf das Control-File wird gerade zugegriffen
           echo bitte Aufruf wiederholen
           exit;;
      121) echo es sind noch User aktiv
           exit;;
      146) echo es wurde keine Sicherung angemeldet
           exit;;
        *) echo Steuerdatei inkonsistent
           echo Systemverwalter ansprechen
           exit;;
       esac


##################
# Datensicherung
##################

     echo -n "Bitte eine formatierte Diskette einlegen.
                                Start der Sicherung = RETURN"
     read a
     tar cfk /dev/rfd0 1300 /daten                             ########
```

```
        *) echo Control-File inkonsistent
           echo Systemverwalter ansprechen
           exit;;
    esac


#################
# Netz abmelden
#################

    # Netz aktiv ??

    a=`/etc/netstat 2>/dev/null`
    if [ $a ]
        then
            echo Netz noch aktiv
            exit
    fi

    # Netz abmelden

    /bin/stopnet 2>/dev/null

    if [ $? -ne 0 ]
        then
            echo stopnet nicht durchführbar
            exit
    fi


######################
# Filesystem abmelden
######################

    /etc/umount /dev/data >/dev/null                    ########
    if [ $? -ne 0 ]
        then
            echo das Filesystem läßt sich nicht abmelden
            /bin/startnet 1>/dev/null 2>/dev/null
            if [ $? -ne 0 ]
            then
                echo startnet nicht durchführbar
            fi
            exit
    fi
```


65

Seite 65 fehlt

```
     if [ $? -ne 0 ]
          then
               echo Datensicherung mit Fehlern abgebrochen
          exit
     fi


#########################################################################
# Zurücksetzen der Sicherungsanforderung im lokalen Control-File
#########################################################################

     resetlock /usr/adm/DSkomm08 2>/dev/null                    ########
     a=$?
     if [ $a -ne 0 ]
          then
               echo "CONTROL-FILE inkonsistent"
          else
               echo "Datensicherung korrekt beendet"
               set `resetlock -v /usr/adm/DSkomm08 2>/dev/null`########
               eval VERSNR=$"$#"
               echo auf die Diskette wurde die Version $VERSNR gesichert
     fi
```

18.11.6.2 Beispiel eines Shell-Script für physiche Datensicherungsicherung

```
G=/../ST008/usr/adm/DSkomm                                      ########
L8=/../ST008/usr/adm/DSkomm08                                   ########
L25=/../ST025/usr/adm/DSkomm25                                  ########

#########################################################
# prüfen, ob die Sicherung durchgeführt werden kann
#########################################################

     testalock $L8 2>/dev/null
     a=$?
     case $a in
          0) ;;
          1) echo auf das Control-File wird gerade zugegriffen
             echo bitte Aufruf wiederholen
             exit ;;
        121) echo "es sind noch User aktiv"
             exit;;
        146) echo "es wurde keine Sicherung angemeldet"
             exit;;
```

```
########################
# Datenträger anmelden
########################

     stctrl -l /dev/rst0 1>/dev/null
     a=$?
     if [ $a -ne 0 ]
          then
          echo Streamer nicht ladbar !
          /etc/mount /dev/data /daten 2>/dev/null                    ########
          if [ $? -ne 0 ]
               then
               echo Filesystem läßt sich nicht anmelden
          fi
          /bin/startnet 1>/dev/null 2>/dev/null
          if [ $? -ne 0 ]
               then
               echo startnet nicht durchführbar
          fi
          exit
     fi


##################
# Datensicherung
##################

     echo "Datensicherung hat begonnen"
     /etc/dicopy /dev/rdata /dev/rst0 1>/tmp/DASIFEH              ########
     if [ $? -eq 0 ]
          then
               echo Datensicherung fehlerhaft
               cat /tmp/DASIFEH
          else
```

```
#########################################################################
# Zurücksetzen der Sicherungsanforderung im LOCAL-Control-FIlE
#########################################################################

     resetlock $L8 2)/dev/null
     a=$?
     if [ $a -ne 0 ]
          then
               echo LOCAL-CONTROL-FILE inkonsistent
          else
               echo Datensicherung korrekt beendet
               set `resetlock -v $L8 2)/dev/null`
               eval VERSNR=$"$#"
               echo auf der Diskette wurde die Version $VERSNR gesichert
          fi
     fi


########################
# Datenträger entladen
########################

     stctrl -u /dev/rst0 1)/dev/null


######################
# Filesystem mounten
######################

     /etc/mount /dev/data /daten 2)/dev/null                    ########
     if [ $? -ne 0 ]
          then
               echo Filesystem läßt sich nicht anmelden
     fi


################
# Netz starten
################

     /bin/startnet 1)/dev/null 2)/dev/null
     if [ $? -ne 0 ]
          then
               echo startnet nicht durchführbar
     fi
```

18.11.6.3 Beispiel für ein C-Programm mit tstaent

```
#include <dasi.h>
#include <stdio.h>

main(argc, argv)
int argc;
char **argv;
{
if (argc != 2)
     {
     printf("%s\n\n","Usage: tst Control-File");
     exit(99);
     }
switch (tstaent(argv[1]))
     {
     case 0:
          exit(0);
     case NOACCESS:
          printf("Kein Zugriff auf Control-File möglich\n");
          exit(NOACCESS);
     case ILLCTLFILE:
          printf("Ungültiges Control-File\n");
          exit(ILLCTLFILE);
     case NOUSERMOD:
          printf("Datensicherung angemeldet\n");
          exit(NOUSERMOD);
     case TOOMANY:
          printf("Kein User-Eintrag mehr frei\n");
          exit(TOOMANY);
     case NOPASSWD:
          printf("User-Eintrag nicht vorhanden (/etc/passwd)\n");
          exit(NOPASSWD);
     case -1:
          printf("system error\n");
          exit(-1);
     default:
          printf("system error\n");
          exit(-1);
     }
}
```

18.11.6.4 Beispiel für ein C-Programm mit leavesys

```
#include <dasi.h>
#include <stdio.h>

main(argc, argv)
int argc;
char **argv;
{
if (argc != 2)
     {
     printf("%s\n\n","Usage: lea Control-File");
     exit(99);
     }
switch (leavesys(argv[1]))
     {
     case 0:
          exit(0);
     case NOACCESS:
          printf("Kein Zugriff auf Control-File möglich\n");
          exit(NOACCESS);
     case ILLCTLFILE:
          printf("Ungültiges Controlfile\n");
          exit(ILLCTLFILE);
     case NOUSERMOD:
          printf("Datensicherung angemeldet\n");
          exit(NOUSERMOD);
     case NOENTRY:
          printf("Kein User-Eintrag im Controlfile vorhanden\n");
          exit(NOENTRY);
     case NOTFOUND:
          printf("User-Eintrag im Control-File nicht gefunden\n");
          exit(NOTFOUND);
     case NOPASSWD:
          printf("User-Eintrag nicht vorhanden (/etc/passwd)\n");
          exit(NOPASSWD);
     case -1:
          printf("system error\n");
          exit(-1);
     default:
          printf("system error\n");
          exit(-1);
     }
}
```