
1 Einleitung

NFS (Network File System) ermöglicht einen verteilten Dateizugriff in Netzen mit unterschiedlichen Rechnerarchitekturen und Betriebssystemen. Voraussetzung für den Einsatz von NFS ist ein Rechnernetz, in dem über TCP/IP kommuniziert werden kann.

1.1 Kurzbeschreibung des Produkts

Die vorliegende Version von NFS für BS2000/OSD entspricht der in Reliant UNIX V5.44 enthaltenen NFS-Implementierung bzw. der AT&T System V Release 4 Variante.

Mit NFS im BS2000 können Sie Datenbestände bearbeiten, die sich an fernen Rechnern befinden, und BS2000-Datenbestände für Benutzer an fernen Rechnern zugänglich machen. Ferne Rechner mit geringer Plattenspeicherkapazität oder PCs können so die erheblich größeren Speicherkapazitäten und weitaus komfortableren und zuverlässigeren Datensicherungsmechanismen von BS2000-Rechnern nutzen.

Voraussetzung für den Einsatz von NFS auf einem BS2000-Rechner ist das Produkt *openNet Server* mit BCAM (ab V15.0).

NFS V1.2C ist ablauffähig ab BS2000/OSD-BC V2.0 und setzt POSIX-BC voraus, das Bestandteil von BS2000/OSD-BC (ab V2.0) ist.

NFS V3.0 ist ablauffähig ab BS2000/OSD-BC V3.0 und setzt POSIX-BC voraus.

1.2 Zielgruppen des Handbuchs

Das Handbuch richtet sich an alle NFS-Anwender. Die Arbeit mit diesem Handbuch setzt voraus, dass Sie über Kenntnisse der Betriebssysteme UNIX und BS2000 verfügen und das Handbuch „POSIX-Grundlagen“ vorliegen haben.

1.3 Wegweiser durch das Handbuch

Kapitel 2 und 3 geben einen Überblick über NFS, seine Einbettung im BS2000 und das Arbeiten mit NFS.

Kapitel 4 enthält die Beschreibung der Kommandos, Dämonen und der Verwaltungsdateien.

Kapitel 5 beschreibt, wie Sie einen PC anbinden und wie Sie die Druckdienste eines BS2000-Rechners vom PC aus nutzen.

Im Kapitel 6 finden Sie Ratschläge zur Fehlerdiagnose und zur Fehlerbehebung sowie Maßnahmen, die Sie zur Leistungsverbesserung von NFS durchführen können.

Das Handbuch enthält ein Fachwort-, Literatur- und Stichwortverzeichnis.

1.4 Readme-Datei

Funktionelle Änderungen und Nachträge der aktuellen Produktversion zu diesem Handbuch entnehmen Sie bitte ggf. der produktspezifischen Readme-Datei. Sie finden die Readme-Datei auf Ihrem BS2000-Rechner unter dem Dateinamen `SYSRME.produkt.version.sprache`. Die Benutzerkennung, unter der sich die Readme-Datei befindet, erfragen Sie bitte bei Ihrer zuständigen Systembetreuung. Die Readme-Datei können Sie mit dem Kommando `/SHOW-FILE` oder mit einem Editor ansehen oder auf einem Standarddrucker mit folgendem Kommando ausdrucken:

```
/PRINT-DOCUMENT dateiname , LINE-SPACING=*BY-EBCDIC-CONTROL
```

bei SPOOL-Versionen kleiner 3.0A:

```
/PRINT-FILE FILE-NAME=dateiname , LAYOUT-CONTROL=  
PARAMETERS(CONTROL-CHARACTERS=EBCDIC)
```

1.5 Darstellungsmittel

In diesem Handbuch werden folgende Darstellungsmittel verwendet:

Im Fließtext

kursive Schrift

alle Elemente der Syntax sowie sonstige Dateinamen, Pfadnamen und Kommandos sind in *kursiver* Schrift dargestellt.



Dieses Symbol kennzeichnet wichtige Hinweise und Ausnahmen, die Sie beachten sollten.

In der Syntax

normale Schrift

Variablen: Diese Zeichen sind Stellvertreter für andere Zeichen, die Sie auswählen und eingeben.

halbfette Schrift

Konstanten: Diese Zeichen sind direkt so einzugeben, wie sie gedruckt sind.

- [] Optional: Alles, was zwischen den eckigen Klammern steht, können, aber müssen Sie nicht eingeben. Die eckigen Klammern selbst dürfen Sie nicht eingeben, es sei denn, es ist ausdrücklich angegeben.
- _ Ein Leerzeichen, das Sie eingeben müssen.
- ... Der vorhergehende Ausdruck kann wiederholt werden. Falls zwischen den Wiederholungen Leerzeichen eingegeben werden müssen, die nicht im Ausdruck enthalten sind, steht vor den Punkten ein Leerzeichen (_).
- | Der senkrechte Strich trennt Alternativangaben.

In Beispielen

halbfette Schreibmaschinenschrift

Eingaben: Eingabezeilen werden bei Zeichenterminals mit der RETURN-Taste, bei Blockterminals mit der Tastenfolge EM DUE abgeschlossen, die Tastenangaben entfallen deshalb.

normale Schreibmaschinenschrift

Ausgaben

1.6 Änderungen gegenüber Vorgängerhandbuch

NFS V3.0 enthält gegenüber der Vorgängerversion folgende Änderungen:

- NFS V3.0 unterstützt zusätzlich neben der NFS-Protokollversion 2 auch die NFS-Protokollversion 3.
- NFS V3.0 unterstützt die volle Funktionalität zum Sperren von Dateien. Es stehen die Dämonen *lockd-srv*, *lockd-clnt* und der Status-Monitor *statd* zur Verfügung.
- NFS V3.0 unterstützt 64-Bit-Systeme. Damit können auch Dateisysteme eingehängt werden, die Dateien enthalten, die größer als 2 GB sind.
- NFS V3.0 unterstützt den asynchronen Schreibmodus (*commit*).

2 Überblick und Einbettung im BS2000

In diesem Kapitel erhalten Sie einen Überblick über NFS. Sie erfahren, wie NFS ins BS2000 eingebettet ist und welche Sicherheitsmechanismen Sie für den Schutz von fernen Ressourcen benutzen können.

2.1 Überblick über NFS

In diesem Abschnitt erhalten Sie grundlegende Informationen über die Funktionsweise von NFS. Zunächst ist die Arbeit mit verteilten Dateisystemen beschrieben. Danach lernen Sie die Unterschiede der verschiedenen NFS-Versionen kennen. Sie erfahren, wie NFS Schreib- und Lesezugriffe auf Dateien ausführt, wie in NFS der Umgang mit Dateisperren realisiert ist und wie mit großen Dateisystemen umgegangen wird.

2.1.1 Arbeiten mit verteilten Dateisystemen

Mit NFS können Sie lokale Dateien und Dateiverzeichnisse für die Bearbeitung an einem fernen Rechner bereitstellen und von fernen Rechnern bereitgestellte Dateien und Dateiverzeichnisse an Ihrem Rechner bearbeiten, als ob es sich um lokale handelt. In Bild 1 kann zum Beispiel das Dateiverzeichnis V1 des BS2000-Rechners am UNIX-Rechner bearbeitet werden, als ob es ein Teil der lokalen Datei-Hierarchie wäre. Selbstverständlich ist auch der umgekehrte Fall möglich, wobei am BS2000-Rechner ein von einem fernen System bereitgestelltes Dateiverzeichnis eingehängt und bearbeitet wird.

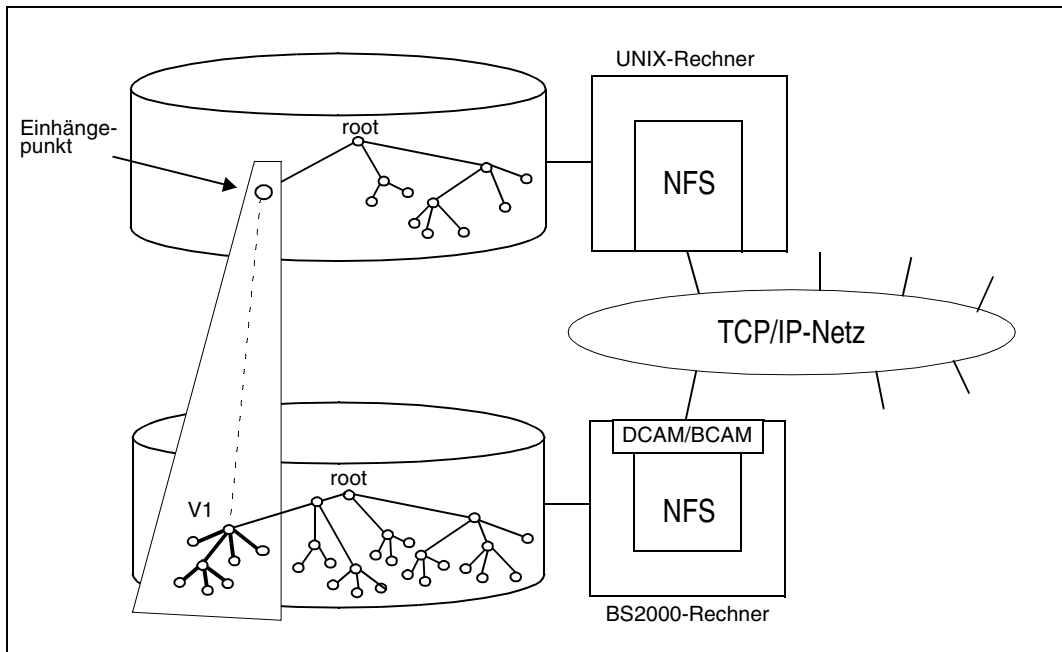


Bild 1: Verteilter Dateizugriff mit NFS in einem heterogenen Netz

Dateisystem, Dateiverzeichnis und Datei

NFS bearbeitet hierarchische Dateisysteme wie z.B. das UNIX-Dateisystem, das aus Dateiverzeichnissen und Dateien besteht. Die gesamte Datei-Hierarchie eines UNIX-Rechners besteht meistens aus mehreren Dateisystemen. Die Dateien und Dateiverzeichnisse eines einzelnen Dateisystems befinden sich physikalisch auf demselben Speichermedium, z.B. auf einer Partition.

Die Dateien und Dateiverzeichnisse aller Dateisysteme eines Rechners bilden zusammen einen Dateibaum. Die Wurzel ist das Root-Dateiverzeichnis. Von hier aus wird in andere Dateiverzeichnisse verzweigt. Dateiverzeichnisse dienen der Gruppierung von Dateien. Die Dateien sind die Blätter des Dateibaums. Von einer Datei aus ist keine Verzweigung mehr möglich.

Ressourcen

Da NFS ein abstraktes Dateisystem-Modell definiert und betriebsystemübergreifend arbeitet, wird für alle Dateien und Datei-Hierarchien, die mit NFS benutzt werden, der Begriff Ressource verwendet. In UNIX-Umgebungen sind Ressourcen beliebige Ausschnitte aus der Datei-Hierarchie, d.h. einzelne Dateien, Dateiverzeichnisse oder ein Dateisystem.

Client und Server

NFS basiert auf dem Client/Server-Prinzip:

- Ein System wird als NFS-Server bezeichnet, wenn es lokale Ressourcen für ferne Systeme bereitstellt.
- Ein System wird als NFS-Client bezeichnet, wenn es Ressourcen benutzt, die von einem fernen System bereitgestellt wurden.

BS2000 kann, wie jedes System mit einem lokalen Massenspeicher, gleichzeitig Client und Server sein, wenn es sowohl auf die Ressourcen anderer Systeme zugreift als auch lokale Ressourcen für andere Systeme bereitstellt.

Bereitstellen (*share*) und Einhängen (*mount*)

Ein NFS-Server stellt seine lokale Datei-Hierarchie oder Ausschnitte daraus mit dem Kommando *share* für die Bearbeitung durch NFS-Clients in anderen Systemen bereit. Der NFS-Server kann beliebige Teile seiner Datei-Hierarchie bereitstellen. Die Teile dürfen sich allerdings nicht überlappen.

Ein NFS-Client, der von einem fernen System bereitgestellte Ressourcen bearbeiten will, muss die fernen Ressourcen in seiner Datei-Hierarchie mit dem Kommando *mount* einhängen und kann dann auf sie zugreifen, als ob sie zu seiner lokalen Datei-Hierarchie gehören. Der Name des Verzeichnisses, unter dem die ferne Ressource eingehängt ist, wird Einhängepunkt genannt. Wenn in diesem Verzeichnis bereits Dateien und Dateiverzeichnisse vorhanden sind, werden sie durch die eingehängte ferne Ressource verdeckt.

Transparenter Zugriff

NFS-Clients hängen die fernen Ressourcen in ihren lokalen Dateisystemen ein. Beim Einhängen wird keine Kopie der fernen Ressource erstellt. Die Bearbeitung der fernen Ressource erfolgt über eine Reihe von Prozeduraufrufen (RPCs), die dem Benutzer des lokalen Systems jedoch verborgen bleiben. Für den Benutzer ist kein Unterschied zwischen lokalen und fernen Ressourcen erkennbar. Er hat nur mit einer einzigen Datei-Hierarchie zu tun, in der er alle Dateien und Dateiverzeichnisse auf dieselbe Art bearbeiten kann.

Eigentümer und Verwalter der „physikalischen“ Dateien ist das System, das die Dateien für ferne Systeme mit dem Kommando *share* bereitstellt, also der jeweilige NFS-Server.

2.1.2 NFS-Versionen und NFS-Protokollversionen

Die Produkt-Version NFS V1.2C unterstützt die NFS-Protokollversion 2. In der Produkt-Version NFS V3.0 können gleichzeitig die NFS-Protokollversionen 2 und 3 verwendet werden. Beim Verbindungsaufbau von Client und Server wird zwischen den Rechnern vereinbart, welche NFS-Protokollversion verwendet werden kann (siehe dazu auch Parameter `v2` des Kommandos *mount*).

Alle NFS-Versionen basieren auf dem UDP-Protokoll (User Datagram Protocol). Dies ist ein verbindungsloses (unreliable) Protokoll, d.h. es werden Datenpakete verschickt, ohne dass von der Transportschicht überprüft wird, ob sie auch den Zielrechner erreichen. Eventuell auftretende Kommunikationsprobleme werden von einer höheren Netzwerkschicht, d.h. dem NFS selbst reguliert. Nach einer bestimmte Zeit ohne Reaktion des Servers wird dann vom Client der Auftrag wiederholt und gegebenenfalls eine Fehlermeldung ausgegeben. Hierdurch wurde um den Preis einer zusätzlichen Belastung des Netzwerks eine sichere Übertragung der Daten gewährleistet.

Mit der NFS-Protokollversion 3 wird im Vergleich zur Protokollversion 2 vor allem wegen der Einsparung von Aufrufen zur Ermittlung von Dateiattributen eine bessere Performance erzielt, ohne die Zuverlässigkeit für den Netzzugriff einzuschränken.

Außerdem können mit NFS V3.0 auch Dateisysteme eingehängt werden, die Dateien enthalten, die größer als 2 GB sind. Um mit solchen Dateien arbeiten zu können, wurde von X/Open eine Applikationsschnittstelle beschrieben, die den Übergang zu den großen Dateien ermöglicht und die mit CRTE V2.2B unterstützt wird (siehe auch Abschnitt „Dateisystemgrößen ab 2 Gbyte“ auf Seite 12).

2.1.3 Der Schreib- und Lesevorgang bei NFS

Der Schreibvorgang

Die Geschwindigkeit, mit der eine Applikation ihre Schreibvorgänge durchführen kann, wird an der Zeitdauer gemessen, die gebraucht wird, bis sich die Daten auf einem sicheren Datenträger, d.h. normalerweise auf einer Festplatte befinden.

Die NFS-Protokollversion 2 verwendete den sogenannten synchronen Schreibmodus (synchronous write). Dieses Verfahren besteht darin, dass eine Applikation auf einem Client einen Schreibvorgang in seinen lokalen NFS-Cache im Hauptspeicher schreibt und diesen an den jeweiligen Server weiterleitet. Hierbei wird jeder Schreibzugriff der Applikation in einen Schreibzugriff auf den NFS-Server umgesetzt. Der Server empfängt die Schreibwünsche und führt diese auf seiner Festplatte aus. Anschließend meldet er dem Client die Ausführung, damit der Client die Daten aus seinem Cache löschen und die Applikation den Schreibvorgang positiv abschließen kann. Meldet der Server nach einer bestimmten Zeit nicht die Erledigung, so überträgt der Client die Daten aus dem Cache einfach noch einmal.

Die NFS-Protokollversion 3 verwendet den sogenannten asynchronen Schreibmodus (safe asynchronous write). Hier werden wie bei der NFS-Protokollversion 2 die Schreibzugriffe auf dem Client im Cache gespeichert und zum Server übertragen. Der Server kann aber dem Client sofort eine Empfangsbestätigung zurückgeben, ohne den Schreibzugriff tatsächlich ausführen zu müssen. Für die Applikation auf dem Client gilt dies als erfolgter Schreibvorgang, so dass diese weiterarbeiten kann. Die Daten verbleiben zur Sicherheit zunächst im NFS-Cache des Clients. Der Server kann mehrere Schreibvorgänge von verschiedenen Clients in seinem Cache sammeln und anschließend gemeinsam als einen Schreibvorgang effizient auf die Festplatte schreiben. Später wird vom NFS-System des Clients eine Abfrage zur Bestätigung des Schreibvorgangs verlangt (commit). Dabei wird nicht für jeden Schreibvorgang eine eigene Bestätigung benötigt. Sobald die Daten auf der Festplatte stehen, wird diese Abfrage vom Server positiv beantwortet. Der Client bzw. die Applikation kann eine Datei schießen und die entsprechenden NFS-Pakete aus dem Cache löschen, sobald dafür eine entsprechende Bestätigung vorliegt (close-to-open). Hat der Server ein Problem, so kann der Client die Aufträge aus seinem Cache nochmals übertragen bzw., wenn das Problem nicht lösbar ist (beispielsweise nicht genug Speicherplatz vorhanden ist), kann eine entsprechende Fehlermeldung der Applikation erzwungen werden, bevor diese beendet bzw. die Datei geschlossen werden kann.

Lesevorgang

Die Lesegeschwindigkeit wird als Zeitraum definiert, den eine Applikation warten muss, bis diese mit den gewünschten Daten versorgt werden kann. Dabei werden von den NFS-Protokollversionen 2 und 3 die gleichen Verfahren verwendet. Diese bestehen zum einen aus einer Pufferung von Lesedaten im Hauptspeicher des jeweiligen Clients und zum anderen in einem speziellen Leseverfahren, das vorausschauend mehr Daten in den Cache liest als eigentlich benötigt werden (read ahead) Damit soll versucht werden, den nächsten Lesezugriff der Applikation vielleicht bereits mit erledigen zu können.

2.1.4 Der Network Lock Manager

Prozesse, die auf gemeinsam benutzten Dateien arbeiten, synchronisieren den Zugriff auf solche Dateien über Sperren. Sperren auf Dateien oder Dateibereiche (Records) werden mit dem Systemaufruf *fcntl()* gesetzt, freigegeben oder abgefragt. Das NFS-Protokoll unterstützt das Setzen von Sperren nicht, weil das NFS-Protokoll „zustandslos“ angelegt ist: Wenn der NFS-Client beim Server eine Aktion auslöst, erzeugt diese Aktion beim Server keinen Zustand, der mit dem Absturz des Servers verloren wäre. Es wird also entweder

- überhaupt kein Zustand auf dem Server erzeugt (z. B. beim Lesen von Daten) oder
- ein „permanent“ Zustand auf dem Server erzeugt (z. B. beim Erzeugen oder Löschen von Dateien). Permanente Zustände bestehen nach einem Absturz des Servers weiter.

Aus der Sicht eines NFS-Clients ist ein Server, der nach einem Absturz wiederanläuft, nur ein sehr langsam arbeitender Server. Analog ist ein nach Absturz wiederanlaufender Client für den NFS-Server lediglich einer, von dem längere Zeit keine Aufträge gekommen sind. Dieses komfortable Wiederanlaufverhalten macht NFS sehr robust und ist der Zustandslosigkeit des NFS-Protokolls zu verdanken.

Das Setzen von Sperren ist mit einem zustandslosen Protokoll aber nicht möglich. Sperren müssen beim NFS-Server registriert werden und sind dort nach einem Server-Absturz verloren (es sei denn, man legt jede Sperre in einem nichtflüchtigen Speicher ab). Das Arbeiten mit Sperren auf einem NFS wird daher in einem gesonderten Protokoll „NLM“ (Network Lock Manager) realisiert. Vom Network Lock Manager werden nur sogenannte „advisory“ Sperren unterstützt. D.h. die Sperren wirken nur, wenn alle Benutzerprogramme für die Zugriffe auf die entsprechenden Dateien bzw. Dateiabschnitte mit Sperren arbeiten. Das Programm *lockd-srv* implementiert das NLM-Protokoll auf der Server-Seite. Beim Client ist NLM im Systemkern implementiert. Der Client ist darüber hinaus aber noch auf Hilfsdienste angewiesen, die ihm der auf dem Client laufende Prozess *lockd-clnt* zur Verfügung stellt.

Beim Arbeiten mit Sperren über NFS muss unbedingt beachtet werden, dass die Client-Prozesse jeden Zugriff auf eine gemeinsam benutzte Datei puffern. Das Setzen von NFS-Sperren sorgt dafür, dass der NFS-Client die Pufferung von Lese- und Schreibdaten unterlässt. Wenn aber sowohl ungepufferte als auch gepufferte Zugriffe benutzt werden (d. h. solche, vor denen keine Sperre gesetzt wurde), kann die Konsistenz der gemeinsam benutzten Dateien nicht mehr garantiert werden. Ausnahme: Alle Client-Prozesse arbeiten auf demselben Client-Rechner (d. h. sie benutzen auch den Puffer gemeinsam).

Fällt ein NLM-Client oder -Server aus, dann müssen bei seinem Wiederanlauf beim jeweiligen Partner Aufräumaktionen gestartet werden. Ein Server muss alle Sperren eines wiederanlaufenden Clients löschen. Ein Client muss seine Sperren beim wiederanlaufenden Server erneut setzen, um den Ausgangszustand vor dem Server-Absturz wiederherzustellen.

Wiederanlaufaktionen können für beliebige zustandsbehaftete Netzdienste erforderlich sein. Sie werden daher bei einem von NLM unabhängigen Dienst, dem NSM (Network Status Monitor) registriert. Der NSM sorgt dann dafür, dass sie beim Wiederanlauf eines abgestürzten Partners ausgeführt werden. Das NSM-Protokoll ist im Programm *statd* implementiert. Der NLM (*lockd-srv* / *lockd-clnt*) ist bislang der einzige Dienst, der den NSM (*statd*) in Anspruch nimmt (siehe auch die Beschreibung zu *lockd-srv* und *lockd-clnt* im Abschnitt „Dämonen“ auf Seite 64).

2.1.5 Der Status-Monitor

Der Status-Monitor, (als *statd* implementiert) ist so allgemein gehalten, dass er auch andere zustandsorientierte Netzdienste und Anwendungen unterstützen kann. Die Wiederherstellung verlorengegangener Statusinformationen nach einem Systemabsturz ist normalerweise einer der schwierigsten Aspekte bei der Entwicklung von Netz-Anwendungen. Durch den Status-Monitor wird daraus mehr oder weniger eine Routineaufgabe.

Der Status-Monitor dient als zentrale Sammelstelle für Netz-Zustandsinformationen. Er ist als Dämon-Prozess implementiert und verwendet ein einfaches Protokoll, über das Anwendungen den Status anderer Systeme ohne großen Aufwand abrufen können. Sein Einsatz macht das Netz weniger anfällig für Störungen und hilft, Situationen zu vermeiden, in denen sich Anwendungen auf unterschiedlichen Systemen (oder sogar auf dem gleichen System) über den Zustand eines Rechners nicht einig sind. Solche Situationen führen bei vielen Anwendungen zu Inkonsistenzen.

Um vom Status-Monitor über Veränderungen des Netzzustands informiert zu werden, muss eine Anwendung alle Systeme bei dem Monitor registrieren, die dieser überwachen soll. Stürzt eines dieser Systeme ab (oder, um genauer zu sein, wenn eines dieser Systeme nach einem Absturz wieder gestartet wird), informiert der Status-Monitor alle Anwendungen, die dieses System bei ihm registriert haben, über den Neustart. Diese können dann mit entsprechenden Maßnahmen ihre Zustandsinformationen aktualisieren.

Dieser Ansatz bietet folgende Vorteile:

- Der aus der Zusammenarbeit mit dem Status-Monitor resultierende Aufwand hinsichtlich Zeit und Code muss nur bei solchen Anwendungen in Kauf genommen werden, die zustandsorientierte Netzdienste in Anspruch nehmen.
- Die Implementierung zustandsorientierter Netz-Anwendungen wird vereinfacht, da der Status-Monitor Anwendungsentwickler von der Komplexität des Netzes abschirmt.

Weitere Informationen zu dem Status-Monitor finden Sie bei der Beschreibung von *statd* im Abschnitt „Dämonen“ auf Seite 64.

2.1.6 Dateisystemgrößen ab 2 Gbyte

NFS V3.0 unterstützt grundsätzlich Dateien größer als 2 Gbyte (große Dateien) bzw. Dateisysteme größer 2 Gbyte (große Dateisysteme). Mit der kommenden POSIX-Version wird es möglich sein, große Dateisysteme lokal einzurichten und sie NFS-Clients bereitzustellen.

Die Unterstützung großer Dateien erstreckt sich auf alle Zugriffe, bei denen Dateibereiche adressiert werden - dies sind das Lesen, Schreiben, Abfragen und Verändern der Dateigröße sowie das Setzen von Dateisperren.

Zugriffe auf NFS-Dateien werden immer vom NFS-Client initiiert und vom NFS-Server durchgeführt. Sind sowohl Client als auch Server 64-Bit-fähig, können große Dateien uneingeschränkt bearbeitet werden. Problemfälle treten aber auf, wenn Client und Server eine unterschiedliche Wortbreite haben.

Das folgende beschreibt das Verhalten beim Einsatz unterschiedlicher Betriebssystem- und NFS-Versionen. Das hier beschriebene Verhalten kann unter Umständen vom Verhalten von Systemen mit anderen Implementierungen abweichen (z. B. Systeme von Fremdherstellern).

32-Bit-Clients und 64-Bit-Server

Bedient ein 64-Bit-Server 32-Bit-Clients stellt sich das Problem, dass der Client bestimmte Datei- oder Dateisystem-Parameter, z. B. die Größe einer Datei, unter Umständen nicht interpretieren kann. Analog zum Verhalten von 32-Bit-Applikationen auf lokalen 64-Bit-Dateisystemen können Systemaufrufe des Clients in diesen Fällen fehlschlagen.

Ausnahme:

Alle Clients, für die ein großes Dateisystem bereitgestellt ist, dürfen dieses normalerweise auch einhängen, unabhängig davon, ob die Dateisystem-Parameter auf den Clients darstellbar sind oder nicht.

Verhalten gegenüber Clients mit der NFS-Protokollversion 2

Wenn von POSIX ein großes Dateisystem für NFS-Clients bereitgestellt wird, dürfen auch Clients der Protokollversion 2 dieses einhängen. Bis zu einer Dateisystemgröße von 1 Tbyte werden die Dateisystem-Parameter (Anzahl der Blöcke etc.) korrekt angezeigt. Der Server lehnt aber jeden Zugriff des Clients auf eine große Datei ab, da der Client die Dateigröße nicht in 32 Bit darstellen kann.

Verhalten gegenüber Clients mit der NFS-Protokollversion 3

Wenn die NFS-Protokollversion 3 verwendet wird, kann der Server nicht zwischen einem 32-Bit- und einem 64-Bit-Client unterscheiden. Es ist daher Sache des 32-Bit-Clients, zu entscheiden, wie er bei Datei- oder Dateisystem-Parametern verfährt, die in 32 Bit nicht darstellbar sind. Normalerweise wird der Client einen Systemaufruf fehlschlagen lassen, wenn Ergebnisparameter nicht in 32 Bit ausgedrückt werden können.

2.2 NFS im BS2000/OSD

NFS im BS2000/OSD gehört zu den POSIX-Programmpaketen. Im folgenden sind die Voraussetzungen für den Einsatz von NFS im BS2000/OSD und die Zusammenarbeit von NFS und POSIX beschrieben.

2.2.1 Netzanbindung

Siehe hierzu die Handbücher „BCAM, Band1 und Band2“.

Voraussetzung für das Arbeiten mit NFS ist ein Netz, in dem über TCP/IP kommuniziert werden kann. Dazu benötigen Sie das Produkt *openNet Server*. Für den Anschluss eines BS2000-Rechners an ein TCP/IP-Netz wird BCAM (ab V12) benötigt. BCAM ist Bestandteil von *openNet Server*.

Partner von NFS im BS2000/OSD können alle anderen Rechner des Netzes sein, die ebenfalls über einen TCP/IP-Anschluss verfügen und auf denen NFS oder ein kompatibles Softwareprodukt für verteilten Dateizugriff installiert ist.

2.2.2 POSIX

Siehe hierzu das Handbuch „POSIX-Grundlagen“.

Mit POSIX (**P**ortable **O**pen **S**ystem **I**nterface for **U**NIX) steht im BS2000/OSD ab V2.0 eine UNIX-ähnliche Umgebung zur Verfügung, die dem POSIX-Standard bzw. dem XPG4-Standard Band1 und Band2 genügt. POSIX im BS2000 wird in zwei Produkten ausgeliefert:

- POSIX-BC ist Bestandteil des Grundausbaus von BS2000/OSD V2.0 und enthält die POSIX-Shell mit einer Grundausstattung an POSIX-Kommandos (auch Basis-Shell genannt), die C-Bibliotheksfunktionen und das POSIX-Subsystem.
- POSIX-SH enthält zusätzliche POSIX-Kommandos.

Voraussetzung für NFS ist POSIX-BC. Die im folgenden aufgeführten Bestandteile von POSIX-BC werden von NFS oder beim Arbeiten mit NFS benutzt.

POSIX-Shell

Mit dem Programm Shell steht im BS2000 eine UNIX-ähnliche Kommandoschnittstelle zur Verfügung, in der Sie POSIX-Kommandos eingeben und POSIX-Programme starten können.

Sie starten die Shell im BS2000-Kommandomodus mit: `/START-POSIX-SHELL`

Sie beenden die Shell mit dem POSIX-Kommando: `exit`

POSIX-Subsystem

Das POSIX-Subsystem ist ein TPR-Subsystem, das die Aufträge von privilegierten und nicht-privilegierten Benutzern bearbeitet. Es übernimmt im BS2000 die Aufgaben des UNIX-Systemkerns. Es besteht aus:

- einem UNIX-Systemkern, der ins BS2000 portiert wurde
- BS2000-Anschlüssen und Diensten, die eine Verbindung zwischen dem portierten UNIX-Systemkern und dem BS2000 herstellen
- Routinen für die Initialisierung und die Beendigung des POSIX-Subsystems

C-Bibliotheksfunktionen

Mit CRTE ab V2.0 stehen außer den BS2000-C-Bibliotheksfunktionen auch die C-Bibliotheksfunktionen der POSIX-Programmschnittstelle zur Verfügung. Damit können Programme im BS2000 die Funktionalität ausführen, die der POSIX-Standard fordert und z.B. auch POSIX-Dateisysteme bearbeiten.

POSIX-Dateisystem

Mit POSIX ist im BS2000 ein weiteres Dateisystem verfügbar: das POSIX-Dateisystem. Es entspricht einem UNIX-Dateisystem. Mit diesem Dateisystem arbeitet NFS.

Ein POSIX-Dateisystem wird mit dem POSIX-Installationsprogramm angelegt. Es wird in einer BS2000-PAM-Datei (Primary Access Method) abgelegt. Die Verteilung der Dateihierarchie auf mehrere PAM-Dateien im BS2000 entspricht der Verteilung auf Partitions im UNIX. PAM-Dateien, in denen sich ein POSIX-Dateisystem befindet, werden auch Behälterdatei genannt.

Das POSIX-Dateisystem besteht aus Dateiverzeichnissen und Dateien. Die Dateien des POSIX-Dateisystems sind byte-orientiert. POSIX behandelt Dateien standardmäßig im EBCDIC-Format.

2.2.3 Bestandteile von NFS

NFS besteht aus Kommandos und Dämonen und benutzt außerdem einige Verwaltungsdateien.

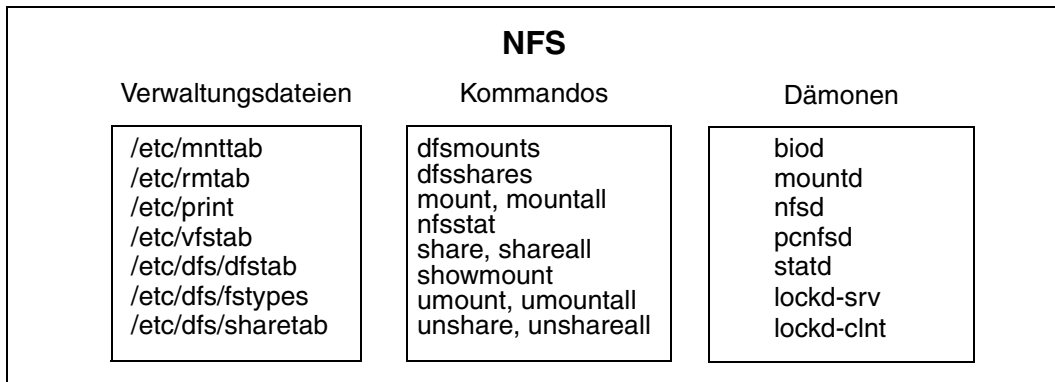


Bild 2: Bestandteile von NFS

Kommandos

NFS bietet Kommandos zum Bereitstellen von lokalen und zum Einhängen von fernen Ressourcen sowie zur Ausgabe von Informationen über bereitgestellte bzw. eingehängte Ressourcen.

Die Kommandos *mount*, *mountall* und *umount*, *umountall* sind in ihrer Grundform in POSIX-BC enthalten. Mit NFS können diese Kommandos mit zusätzlichen Optionen bzw. mit NFS-spezifischer Funktionalität verwendet werden.

Dämonen

Dämonen sind Systemprozesse, die permanent und meist im Hintergrund laufen und die allgemeine Aufgaben durchführen. Die NFS-Dämonen koordinieren die Vorgänge, die übers Netz gehen, wie z.B. das Einhängen und die Ein-/Ausgabe-Aktionen. Außerdem unterstützen sie die PC-Anbindung. Die NFS-Dämonen werden automatisch beim Starten von NFS mitgestartet.

Verwaltungsdateien

Die Verwaltungsdateien unterstützen die Verwaltung von Ressourcen. Sie enthalten entweder Informationen für den Benutzer, die mit Hilfe von Kommandos ausgegeben werden oder Informationen für Kommandos, die entweder der Benutzer oder Kommandos in diese Dateien eingetragen hat.

RPC (Remote Procedure Call)

Zur Netzkommunikation benutzt NFS ferne Prozeduraufrufe (RPCs). Hierfür stehen mit POSIX-BC der Dämon *rpcbind* und das Programm *rpcinfo* zur Verfügung. Der *rpcbind*-Dämon vermittelt die Adressen der Kommunikationspartner. Das *rpcinfo*-Programm gibt Auskunft über die Verbindungen, Transportwege, Programme usw., die bei der Netzkommunikation benutzt werden.

2.2.4 Zusammenwirken von NFS und POSIX

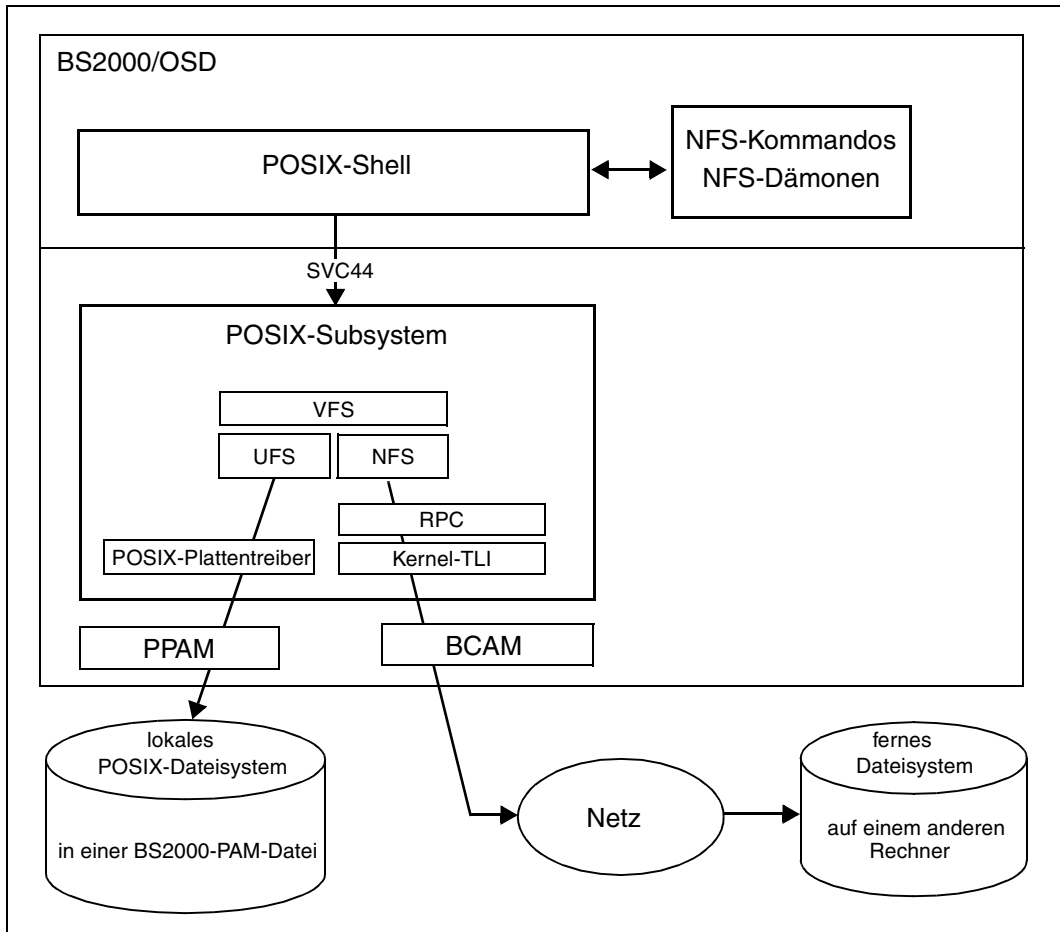


Bild 3: NFS und POSIX (Sicht als Client)

NFS als POSIX-Programmpaket setzt auf POSIX auf und nutzt folgende POSIX-Funktionalität:

- In der POSIX-Shell werden die NFS-Kommandos eingegeben.
- Im POSIX-Subsystem werden u.a. ferne Dateisystem-Anforderungen bearbeitet.

Zum Beispiel wird eine Dateisystem-Anforderung zuerst der POSIX-Subsystemkomponente VFS (Virtual File System) übergeben. Sie erkennt anhand des Dateisystemtyps, ob ein lokales oder ein fernes Dateisystem bearbeitet werden soll:

- Wenn ein lokales Dateisystem bearbeitet werden soll, wird die Anforderung über die POSIX-Komponente UFS (UNIX File System), den POSIX-Plattentreiber und das BS2000-Produkt PPAM (Privileged Primary Access Method) abgewickelt.
- Wenn eine ferne Ressource bearbeitet werden soll, wird die Anforderung über NFS abgewickelt. Über RPCs (Remote Procedure Calls) sowie Kernel-TLI (Transport Layer Interface) und das BS2000-Produkt BCAM wird mit dem fernen Rechner kommuniziert.

2.3 Sicherheit

Mit NFS kann von Rechnern in einem lokalen Netz auf Dateien eines fremden Rechners zugegriffen werden; die Datenbestände im Netz können gemeinsam benutzt werden. Unberechtigte Zugriffe können Sie mit den verschiedenen Schutzmechanismen von BS2000, POSIX und NFS abwehren. Folgende Schutzmechanismen stehen zur Verfügung:

- Benutzerverwaltung (BS2000)
- Dateizugriffsschutz (BS2000, POSIX und NFS)
- Datensicherung (BS2000)
- Portüberwachung (NFS)

2.3.1 Benutzerverwaltung

Siehe hierzu das Handbuch „POSIX-Grundlagen“.

Die BS2000-Benutzerverwaltung SRPM (System Resources and Privileges Management) kontrolliert den Zugang zum BS2000 und zum POSIX. Jeder Benutzer muss im BS2000-Benutzerkatalog eingetragen sein. Für die Verwaltung von BS2000-Benutzern ist der BS2000-Systemverwalter zuständig.

Die POSIX-Benutzerverwaltung ist in die BS2000-Benutzerverwaltung integriert, sie wird nicht, wie in UNIX üblich über die Datei *etc/passwd* abgewickelt. Für die POSIX-Benutzerverwaltung stehen, bei entsprechender Berechtigung, die folgenden SDF-Kommandos zur Verfügung; mit ihnen wird z.B die POSIX-Benutzernummer (uid) vergeben:

```
/ADD-USER  
/MODIFY-USER-ATTRIBUTES  
/SHOW-USER-ATTRIBUTES  
/MODIFY-POSIX-USER-ATTRIBUTES  
/SHOW-POSIX-USER-ATTRIBUTES  
/MODIFY-POSIX-USER-DEFAULTS  
/SHOW-POSIX-USER-DEFAULTS
```

Für die POSIX-Gruppenverwaltung und die Verwaltung von Zugängen über ferne Rechner mit *rlogin* gibt es weitere Verfahren, Kommandos und Operanden, die im obengenannten Handbuch beschrieben sind.

Root-Berechtigung (uid=0 und gid=0)

Ein Benutzer hat die Root-Berechtigung, wenn ihm die POSIX-Benutzernummer (uid) 0 und die POSIX-Gruppennummer (gid) 0 zugeteilt wurde. Die Root-Berechtigung benötigen Sie für den Start von NFS und die POSIX-interne Verwaltung von Dateisystemen. Der Kennung SYSROOT ist standardmäßig die Root-Berechtigung zugeteilt. Der Kennung TSOS wird die Root-Berechtigung bei der Installation von POSIX zugeteilt.

Die Root-Berechtigung hat nur für das lokale System Gültigkeit. Root-Berechtigte von ferneren Systemen sind im lokalen System nicht-privilegierten Benutzern gleichgestellt. Wenn ein ferner Root-Berechtigter unter der Benutzernummer 0 arbeiten soll, kann ihm mit Optionen des Kommandos *share* die entsprechende Berechtigung erteilt werden:

```
share -F nfs -o ...,root=fernes-system ...
```

Kennung TSOS

Zusätzlich zur Root-Berechtigung benötigen Sie das BS2000-Privileg TSOS, gebunden an die BS2000-Kennung TSOS, für alle Verwaltungsaufgaben, bei denen Sie BS2000-Dateien fremder Kennungen verwalten müssen, z.B. für das Anlegen von Behälterdateien, in denen Sie neu einzurichtende Dateisysteme ablegen wollen.

2.3.2 Dateizugriffsschutz

Siehe hierzu das Handbuch „POSIX-Grundlagen“.

POSIX-Dateien und POSIX-Dateiverzeichnisse können innerhalb von POSIX durch Schutzbits vor unberechtigtem Zugriff geschützt werden. Die Behälterdateien, die die einzelnen Dateisysteme des POSIX-Dateibaums enthalten, können im BS2000 zusätzlich durch entsprechende Dateiattribute geschützt werden. Für verteilte Ressourcen können Sie zusätzlich bei der Bereitstellung bzw. beim Einhängen die Zugriffe regeln.

Zugriffsschutz für Behälterdateien

Das POSIX-Installationsprogramm legt Behälterdateien mit den Attributen USER-ACCESS=*OWN und ACCESS=*WRITE für die angegebene Kennung an. Diese Attribute sollten Sie nicht verändern. Außerdem dürfen Sie kein Dateikennwort vergeben.

Der Benutzer einer POSIX-Datei benötigt kein Zugriffsrecht für die Behälterdatei, in der die POSIX-Datei liegt.

Schutzbits

Der Zugriffsschutz für Dateien und Dateiverzeichnisse ist in POSIX durch Schutzbits realisiert, wie sie in UNIX üblich sind. Es gibt drei Schutzbits, die für jede der drei Benutzerklassen (Eigentümer, Gruppe, Andere) individuell vergeben werden können, und ein Identifikationszeichen.

Identifikationszeichen	Eigentümer	Gruppe	Andere
- Datei	r read	r read	r read
b blockorientiertes Gerät	w write	w write	w write
c zeichenorientiertes Gerät	x execute	x execute	x execute
d Dateiverzeichnis			
l symbolischer Verweis			
s Ausführungsbit			

Tabelle 1: Schutzbits und Identifikationszeichen

Beispiel 1

```
- rwx r-- r--
```

Diese Zeichen zeigen an, dass es sich um eine Datei handelt, die der Eigentümer lesen, schreiben und ausführen darf, andere Mitglieder seiner Gruppe und sonstige Benutzer dürfen die Datei nur lesen.

Beispiel 2

```
d rwx r-- r--
```

Diese Zeichen zeigen an, dass es sich um ein Dateiverzeichnis handelt, in dem der Eigentümer Einträge lesen, anlegen/löschen und das er durchsuchen darf, andere Mitglieder seiner Gruppe und sonstige Benutzer dürfen in dem Dateiverzeichnis nur Einträge lesen.

Zugriffsschutz für ferne Ressourcen

Die Kommandos für die Bereitstellung und das Einhängen von Ressourcen bieten Optionen, mit denen der Zugriff von Clients differenziert geregelt werden kann. Im folgenden sind einige Berechtigungen aufgeführt, die Sie allen oder ausgewählten Clients einräumen können:

- die Root-Berechtigung
- den Nur-Lese-Zugriff
- den Lese- und Schreib-Zugriff
- die Erlaubnis zum Anlegen von Gerätedateien bzw. den Zugriff darauf

Durch die Kombination von Benutzerberechtigungen und Dateizugriffs-Schutzmechanismen können Sie für verteilte Ressourcen den jeweils notwendigen Schutz erreichen.

Beispiel

Das Dateiverzeichnis `/usr1/v1` gehört dem Benutzer mit `uid=4712`.

Es hat folgende Schutzbits: `d rwx r-- r--`

Es wird mit dem Kommando `share -F nfs /usr1/v1` für ferne NFS-Clients mit Schreib- und Lesezugriff bereitgestellt. Alle Clients, die diese Ressource bei sich einhängen, können nun lesend auf sie zugreifen. Clients, die schreibend darauf zugreifen wollen, müssen auf ihrem Rechner unter derselben `uid` wie der Eigentümer, also `4712`, angemeldet sein, da die Schutzbits den schreibenden Zugriff von *Gruppe* und *Andere* nicht zulassen.

2.3.3 Portüberwachung

Bei Kommunikation über TCP/IP erfolgt die Adressierung von Anwendungen im Netz über die Kombination Internetadresse/Portnummer, die eindeutig den Empfänger bzw. den Sender eines Datenpaketes identifiziert. Die Internetadresse adressiert den Rechner, die Portnummer adressiert die Anwendung innerhalb des Rechners.

Eine zusätzliche Sicherheitsleistung von NFS ist die Überwachung der Portnummern. Sie ist standardmäßig eingeschaltet. Bei eingeschalteter Portüberwachung überprüft der NFS-Server die Portnummern, an die ein NFS-Client seine Aufträge sendet. Er prüft bei jedem Client-Zugriff, ob die Portnummer über die der Client-Auftrag hereinkommt privilegiert, d.h. kleiner als 1024 ist. Wenn sie nicht privilegiert ist, wird der Client-Auftrag vom Server zurückgewiesen.

Die Portüberwachung können Sie durch den Parameter *PORTMON* in der POSIX-Informationsdatei *SYSSSI.POSIX-BC.040* ein- und ausschalten. Siehe hierzu „POSIX installieren und starten“ auf Seite 26:

PORTMON=1 Portüberwachung ist eingeschaltet; Client-Zugriffe über nicht privilegierte Portnummern werden zurückgewiesen; Standardverhalten

PORTMON=0 Portüberwachung ist ausgeschaltet



In einem BS2000-System können privilegierte Portnummernbereiche auch anders gesetzt sein. NFS betrachtet jedoch immer die Portnummern kleiner als 1024 als privilegiert.

3 Installation und Einsatz

In diesem Kapitel erfahren Sie, wie Sie NFS installieren, wie Sie die POSIX-Shell starten und beenden, wie Sie NFS starten und beenden, wie Sie Dateisysteme bereitstellen und einhängen und was Sie beim Arbeiten mit POSIX-Dateien im BS2000 beachten müssen.

3.1 Installation von NFS

Die Installation von NFS ist im POSIX-Installationsprogramm integriert. Im folgenden ist das Vorgehen bei der Installation beschrieben.

Netzanbindung

Voraussetzung für den Einsatz von NFS ist die Einbindung des BS2000-Rechners in ein TCP/IP-Netz. Hierfür benötigen Sie das Produkt *openNet Server*.

Dateien der Liefereinheit einspielen

Die Dateien der Liefereinheit NFS müssen unter der Standardkennung (\$.) eingespielt werden. Dies erfolgt üblicherweise mit dem Verfahren SOLIS. Ist im Ausgangssystem das Produkt IMON (Installations MONitor) gestartet, kann die Installation über IMON erfolgen. Ab BS2000/OSD-BC V3.0 bzw. OSD-SVP V2.0 muss die Installation mit dem Installationsmonitor IMON durchgeführt werden.

Die Liefereinheit NFS besteht aus folgenden Dateien:

SINLIB.NFS.030 bzw. SINLIB.NFS.012	Bibliothek mit Kommandos, Dämonen etc.
SYSSII.NFS.030 bzw. SYSSII.NFS.012	IMON-Infodatei
SYSFGM.NFS.030.D bzw. SYSFGM.NFS.012.D	Freigabemitteilung in deutsch
SYSFGM.NFS.030.E bzw. SYSFGM.NFS.012.E	Freigabemitteilung in englisch

Tabelle 2: Die Dateien der Liefereinheit NFS

POSIX installieren und starten

Siehe hierzu das Handbuch „POSIX-Grundlagen“.

Berechtigung:

Für das Installieren und Starten von POSIX müssen Sie das BS2000-Privileg TSOS (gebunden an die Kennung TSOS), das BS2000-Privileg SUBSYSTEM-ADMINISTRATION und die Root-Berechtigung besitzen.

Mit POSIX wird ein Installationsprogramm ausgeliefert. Mit diesem Programm können Sie:

- das POSIX-Subsystem installieren (Install POSIX subsystem)
- POSIX-Dateisysteme verwalten (Administrate POSIX filesystems)
- POSIX-Programmpakete installieren (Install packages on POSIX)
- POSIX-Programmpakete de-installieren (Delete packages from POSIX))

Nach der abgeschlossenen Erstinstallation von POSIX ist das POSIX-Subsystem aktiv und das POSIX-Dateisystem eingerichtet. Als minimale Ausstattung enthält ein POSIX-Dateisystem nach der Erstinstallation von POSIX das Root-Dateiverzeichnis / und das Verzeichnis /var, jeweils in einer eigenen Behälterdatei.

Das POSIX-Subsystem kann nun beliebig aktiviert und deaktiviert werden. Hierzu können Sie die folgenden SDF-Kommandos verwenden:

/START-SUBSYSTEM POSIX	aktivieren des POSIX-Subsystems
/STOP-SUBSYSTEM POSIX	deaktivieren des POSIX-Subsystems
/SHOW-SUBSYSTEM POSIX	informieren über das POSIX-Subsystem
/SHOW-POSIX-STATUS	Status des POSIX-Subsystems anzeigen

Für die Installation des POSIX-Programmpakets NFS muss die Erstinstallation von POSIX abgeschlossen sein, und das POSIX-Subsystem muss aktiv sein.

Parameter ändern

Parameter, wie z.B. die Portüberwachung, ändern Sie in der Informationsdatei SYSSSI.POSIX-BC.040. Die geänderten Parameter werden beim nächsten Start des POSIX-Subsystems übernommen.

NFS installieren

Berechtigung: Für die Installation von NFS benötigen Sie das BS2000-Privileg TSOS und die Root-Berechtigung.

NFS installieren Sie mit dem POSIX-Installationsprogramm. Geben Sie folgendes Kommando ein:

► `/START-POSIX-INSTALLATION`

Anschließend erscheint folgendes Hauptmenü:

```
BS2000 POSIX installation program

Please select

Install POSIX subsystem
Administrate POSIX filesystems
Install packages on POSIX
Delete packages from POSIX

Select: MAR + DUE
Help : F1

Finish installation: F2
```

► Wählen Sie das Menü *Install packages on POSIX*

Anschließend erscheint das Folgemenü:

```

Leopold
BS2000 POSIX package installation

IMON support ?      : Y (y) mandatory for official package
                   : (n) private package (SINLIB...)

name of product    : nfs
package of product :                               (optional for certain products)

version of product : 030█ (format Vmm.n or mmn)

correction state  :                               (format aso, optional for IMON support)

installation userid:                               (mandatory for no IMON support)

install: DUE      help: F1      terminate: F2

LTG                                     TAST

```

- ▶ Tragen Sie als Produkt *nfs* und als Version *030* ein. NFS wird installiert. Für die De-Installation benutzen Sie entsprechend das Menü *Delete packages from POSIX*.

Umstieg von vorherigen NFS-Versionen (V1.2B)

- ▶ Deinstallieren Sie das Vorgängerprodukt mit *Delete packages from POSIX*.
- ▶ Installieren Sie anschließend NFS neu wie oben beschrieben.



NFS V3.0 (bzw. NFS V1.2C) setzt die Korrekturstufe A31 von POSIX voraus. Gegebenenfalls muss eine Update-Installation zu POSIX-BC durchgeführt werden.

3.2 POSIX-Shell starten und beenden

Siehe hierzu das Handbuch „POSIX-Grundlagen“.

Die POSIX-Shell benötigen Sie für den Start von NFS, für den Start von einzelnen NFS-Dämonen und für die Eingabe von NFS-Kommandos bzw. zum Aufrufen des Programms *rpcinfo*. Die POSIX-Shell können Sie nur aufrufen, wenn POSIX installiert ist und das POSIX-Subsystem aktiviert ist.

Die POSIX-Shell starten Sie mit dem BS2000-Kommando: `/START-POSIX-SHELL`

Die POSIX-Shell beenden Sie mit dem Shell-Kommando: `exit`

3.3 Einsatz von NFS

Welche Aufgaben beim Einsatz von NFS anfallen, hängt von den Anforderungen Ihrer Organisation und der Rolle Ihres Systems innerhalb des Netzes ab.

Die Verwaltung eines Rechners nach der Konfiguration umfasst folgende Aufgaben:

- Starten und Stoppen von NFS
- Anlegen und Konvertieren von POSIX-Dateisystemen
- Bereitstellen von Ressourcen und Zurücknehmen der Bereitstellung nach Bedarf während einer Server-Sitzung
- Ein- und Aushängen von Ressourcen nach Bedarf während einer Arbeitssitzung
- Ändern von Verwaltungsdateien, wenn die Listen der automatisch bereitzustellenden bzw. einzuhängenden Ressourcen aktualisiert werden sollen
- Informieren über freigegebene und eingehängte Ressourcen
- Lokalisieren und Beseitigen von Problemen beim Betrieb von NFS (siehe Kapitel „Diagnose und Leistungsverbesserung“ auf Seite 99)

3.3.1 NFS starten und beenden

NFS wird automatisch beim Systemstart über ein Startskript gestartet und beim Beenden des Systems über ein Stopskript beendet.

Beim Starten werden alle Ressourcen bereitgestellt, die in der Datei */etc/dfs/dfstab* mit der Option *-F nfs* eingetragen sind (siehe Abschnitt „Verwaltungsdateien“ auf Seite 80).

Bei Bedarf können Sie NFS jedoch auch manuell starten bzw. beenden.

NFS manuell starten

Berechtigung: Zum Starten von NFS benötigen Sie die Root-Berechtigung.

- ▶ Starten Sie die POSIX-Shell mit dem SDF-Kommando:

```
/START-POSIX-SHELL
```

- ▶ In der POSIX-Shell starten Sie NFS mit dem Skript *S20nfs*. Das Skript *S20nfs* befindet sich im Verzeichnis */etc/rc2.d*. Geben Sie ein:

```
/etc/rc2.d/S20nfs start
```

Beim Start von NFS werden alle Dämonen automatisch gestartet. Es erscheint folgende Meldung auf dem Bildschirm:

```
*** starting NFS V3.0 ***
```

Beim Starten werden alle Ressourcen bereitgestellt, die in der Datei */etc/dfs/dfstab* mit der Option *-F nfs* eingetragen sind. Nach dem Start von NFS können Sie zum Einhängen von vordefinierten Ressourcen folgendes Kommando eingeben:

- ▶ `mountall -F nfs`

Mit dem Kommando *mountall* hängen Sie alle Ressourcen vom Typ *nfs* ein, die in der Datei */etc/vfstab* eingetragen sind.

NFS manuell beenden

Berechtigung: Zum Beenden von NFS benötigen Sie die Root-Berechtigung.

- ▶ Sie beenden NFS mit dem Skript `K20nfs`. Das Skript `K20nfs` befindet sich im Verzeichnis `/etc/rc0.d`. Geben Sie in der POSIX-Shell ein:
- ▶ `/etc/rc0.d/K20nfs stop`

Bei der Beendigung von NFS wird die Bereitstellung von Ressourcen zurückgenommen, es werden alle Dateisysteme vom Typ `nfs` ausgehängt und alle Dämonen beendet. Es erscheint folgende Meldung auf dem Bildschirm:

```
*** NFS V3.0 going down ***
```

3.3.2 Besonderheiten des POSIX-Dateisystems

Anlegen von Dateisystemen

Siehe hierzu das Handbuch „*POSIX-Grundlagen*“.

Berechtigung: Für das Anlegen und Verwalten von POSIX-Dateisystemen benötigen Sie das BS2000-Privileg TSOS und die Root-Berechtigung.

Dateisysteme legen Sie mit dem POSIX-Installationsprogramm, Menü: *Administrate POSIX filesystems* an. Nach abgeschlossener Erstinstallation von POSIX enthält der POSIX-Dateibaum als minimale Ausstattung das Root-Dateiverzeichnis `/` und das Verzeichnis `/var`, jeweils in einer eigenen Behälterdatei. Weitere Dateisysteme können Sie bei Bedarf anlegen.

Beim Anlegen eines neuen Dateisystems wird auch die Größe der Behälterdatei festgelegt. Sie kann nachträglich nicht mehr geändert werden.

Speicherung der Dateisysteme

Der gesamte POSIX-Dateibaum ist in einer oder in mehreren Behälterdateien (das sind BS2000-PAM-Dateien) abgespeichert.

Die Behälterdateien liegen auf Platten (PVS, Public Volume Set). Um eine bessere Performance zu erreichen, sollten Sie nicht alle Behälterdateien auf dieselbe Platte legen.

Namenskonventionen

Die Namen von POSIX-Dateien können maximal 1024 Zeichen lang sein. Dabei sind die Namen der Dateiverzeichnisse, der eigentliche Dateiname sowie die begrenzenden Schrägstriche bereits mitgerechnet.

Übernahme von Dateisystemen, die mit NFS V1.0 angelegt wurden

Dateisysteme, die Sie mit der NFS V1.0 angelegt haben, können Sie im POSIX-Dateibaum einhängen. Der Einhängepunkt darf weder `/` (root) noch `/var` sein, d.h. Sie dürfen die bei der Erstinstallation von POSIX angelegten Dateisysteme `/` und `/var` nicht durch die der Vorgängerversion verdecken.

Die Behälterdateien mit den einzuhängenden Dateisystemen der Vorgängerversion müssen im Non-Key-Format vorliegen. Wenn eine Behälterdatei im Key-Format vorliegt, muss sie mit dem BS2000-Dienstprogramm PAMCONV in das Non-Key-Format umgewandelt werden. Danach kann das in ihr enthaltene Dateisystem eingehängt werden.

3.3.2.1 Code-Konvertierung

Siehe hierzu das Handbuch „POSIX-Kommandos“, Kommando: *iconv*

Die Dateien des POSIX-Dateisystems werden von POSIX normalerweise im EBCDIC-Format gemäß EDF03 bzw. EDF04 behandelt. Für die Konvertierung in das ASCII-Format gemäß ISO 646 oder ISO 8859-1 steht das POSIX-Kommando *iconv* zur Verfügung.

Die Codetabellen sind im Verzeichnis */usr/lib/iconv* enthalten.

Für das Bereitstellen von Ressourcen des POSIX-Dateisystems für UNIX- bzw. Windows-Rechner, sollten Sie die Dateien konvertieren. Ebenso ist eine Konvertierung nötig, wenn Sie Dateien im ASCII-Format von fernen Rechnern im POSIX-Dateisystem einhängen wollen.

Wenn Sie ASCII-Dateien automatisch konvertieren lassen wollen, müssen Sie die Shell-Variablen `IO_CONVERSION` auf `YES` setzen. Dies geschieht mit den folgenden Kommandos:

```
IO_CONVERSION=YES
export IO_CONVERSION
```

Beispiel

Der Inhalt der Datei *bs2000* soll von ASCII nach EBCDIC konvertiert werden und das Ergebnis in die Datei *bs2000.conv* geschrieben werden:

```
► iconv -f 646 -t edf04 bs2000 > bs2000.conv
```


3.3.2.2 BS2000-Dateien

Siehe hierzu die Handbücher „POSIX-Grundlagen“ und „POSIX-Kommandos“, Kommando: `bs2cp`

BS2000-Dateien können mit dem POSIX-Kommando `bs2cp` ins POSIX-Dateisystem kopiert werden und so mit NFS für andere Systeme im Netz bereitgestellt werden.

3.3.3 Ressourcen bereitstellen und Bereitstellung zurücknehmen

Berechtigung: Zum Bereitstellen und zum Zurücknehmen der Bereitstellung von Ressourcen benötigen Sie die Root-Berechtigung.

Lokale Ressourcen bereitstellen (exportieren)

Ein NFS-Server stellt lokale Ressourcen für die Bearbeitung durch ferne Systeme (NFS-Clients) bereit. Lokale Ressourcen stellen Sie bereit durch die Kommandos `share` oder `shareall`:

- Mit dem Kommando `share` stellen Sie eine einzelne Ressource für den Client-Zugriff bereit und legen die Zugriffsrechte für Clients fest.
- Mit dem Kommando `shareall` können Sie mehrere Ressourcen gleichzeitig für den Client-Zugriff bereitstellen. Das Kommando erwartet ein oder mehrere `share`-Kommandos in einer benutzerdefinierten Datei, von der Standardeingabe oder in der Verwaltungsdatei `/etc/dfs/dfstab`.

Bereitstellung zurücknehmen

Die Bereitstellung von Ressourcen wird zurückgenommen durch die Kommandos `unshare` oder `unshareall` und automatisch bei der Beendigung von NFS.

Bereitstellen von Ressourcen zur ständigen Benutzung

Wenn Sie anderen Rechnern bestimmte Ressourcen ständig verfügbar machen wollen, können Sie Ihr System so konfigurieren, dass diese Ressourcen beim Start von NFS automatisch bereitgestellt werden. Dieses Verfahren sollten Sie dann wählen, wenn zum Beispiel eine Ressource für andere Rechner ständig verfügbar sein muss und der Client-Zugriff voraussichtlich gar nicht oder nur selten aufgehoben wird.

Die automatisch bereitzustellenden Ressourcen werden in die Datei *dfstab* im Verzeichnis */etc/dfs* eingetragen. Die Datei *dfstab* wird bei der Installation von NFS automatisch angelegt.

Die Datei *dfstab* enthält eine Liste aller Ressourcen, die den Clients nach dem Start von NFS zur Verfügung stehen sollen, sowie die für diese Ressourcen geltenden Zugriffsberechtigungen. Um eine Ressource in *dfstab* einzutragen, einen Eintrag zu löschen oder Optionen zu ändern, editieren Sie die Datei mit einem beliebigen unterstützten Texteditor. Die vorgenommenen Änderungen werden wirksam, sobald das nächste Mal POSIX gestartet wird, ein *shareall*-Kommando ausgeführt wird oder NFS gestoppt und wieder gestartet wird.

3.3.4 Ressourcen ein- und aushängen

Berechtigung: Zum Einhängen und Aushängen von Ressourcen benötigen Sie die Root-Berechtigung.

Ressourcen einhängen (montieren)

Ohne NFS können Sie nur lokale Ressourcen ein- und aushängen, mit NFS können Sie auch alle fernen Ressourcen ein- und aushängen, die von anderen Systemen des Netzes bereitgestellt wurden.

Ein NFS-Client hängt Ressourcen eines fernen Systems (NFS-Servers) in seiner lokalen Dateihierarchie ein. Voraussetzung hierfür ist, dass der NFS-Server verfügbar ist und die gewünschte Ressource bereitgestellt hat. Der NFS-Client muss die Berechtigung zum Zugriff auf diese Ressource haben und der Einhängepunkt muss vorhanden sein.

Falls in dem Verzeichnis, das als Einhängepunkt gewählt wird, lokale Dateien und Dateiverzeichnisse vorhanden sind, werden diese durch die eingehängte ferne Ressource verdeckt.

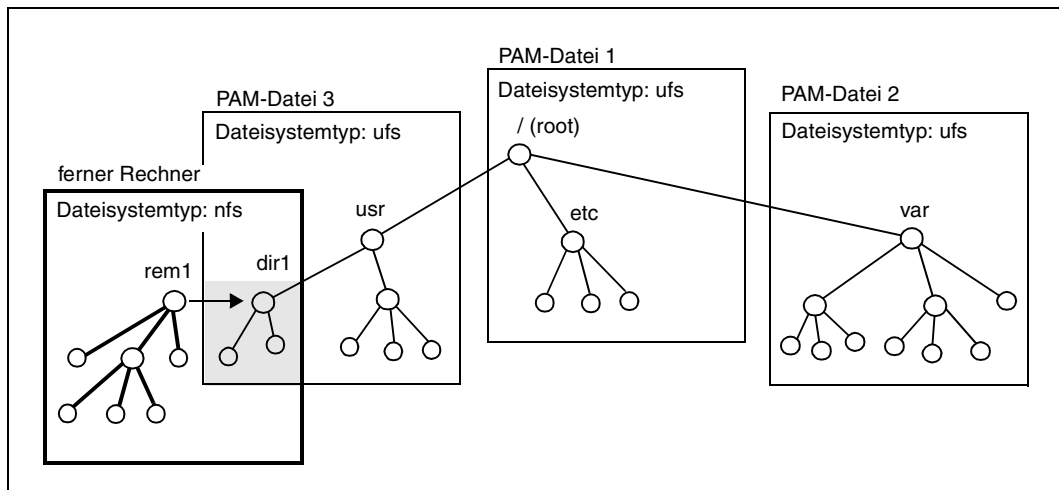


Bild 4: Dateisysteme der POSIX-Dateihierarchie

In Bild 4 besteht die lokale Datei-Hierarchie aus drei POSIX-Dateisystemen und einem entfernten Dateisystem. Als Einhängpunkt für die ferne Ressource *rem1* wird das Verzeichnis *dir1* gewählt. Die beiden lokal in *dir1* vorhandenen Dateien werden durch die ferne Ressource verdeckt.

Ferne Ressourcen hängen Sie ein durch die Kommando *mount* oder *mountall*:

- Mit dem Kommando *mount* hängen Sie eine einzelne Ressource in ihrem lokalen Dateisystem ein und legen die Zugriffsrechte für die Benutzer ihres Systems fest.
- Mit dem Kommando *mountall* können Sie mehrere Ressourcen gleichzeitig einhängen. Das Kommando erwartet Angaben über die einzuhängenden Ressourcen in einer benutzerdefinierten Datei, von der Standardeingabe oder in der Verwaltungsdatei */etc/vfstab*.

Ressourcen aushängen

Ferne Ressourcen werden ausgehängt durch die Kommandos *unmount* oder *unmountall* und automatisch bei der Beendigung von NFS.

Automatisches Einhängen von fernen Ressourcen

Über die Datei */etc/vfstab* können Sie veranlassen, dass ein Dateisystem beim Start von POSIX automatisch eingehängt wird. Wenn Sie in diese Datei ein Dateisystem oder Verzeichnis eines anderen Rechners eintragen, wird die Ressource automatisch beim Start von NFS eingehängt. Bevor Sie eine ferne Ressource automatisch einhängen können, müssen Sie mit dem Kommando *mkdir* einen Einhängepunkt erzeugen und die Datei *vfstab* editieren.

3.3.5 Informieren über Ressourcen

Mit den Kommandos *share*, *mount*, *dfshares*, *dfmounts* und *showmount* können Sie feststellen, auf welche fernen Ressourcen Sie aktuell zugreifen können, welche Ressourcen auf ihrem System eingehängt und für den Client-Zugriff bereitgestellt sind und welche lokalen Ressourcen bei fernen Clients eingehängt sind.

Informieren über bereitgestellte Ressourcen

- Das Kommando *share* ohne Operanden zeigt alle lokalen Ressourcen an, auf die Clients zugreifen dürfen.
- Die Verwaltungsdatei *sharetab* enthält Informationen über alle lokalen Ressourcen, die mit dem Kommando *share* bereitgestellt wurden.
- Das Kommando *dfshares* zeigt alle fernen Ressourcen an, die für Client-Zugriffe bereitgestellt wurden.

Informieren über eingehängte Ressourcen

- Das Kommando *mount* ohne Operanden zeigt alle lokalen und fernen Ressourcen an, die im lokalen Dateisystem eingehängt sind. Die Operanden *-p* oder *-V* geben ebenfalls Informationen über eingehängte Ressourcen. Siehe hierzu das Handbuch „POSIX-Kommandos“, Kommando: *mount*.
- Das Kommando *dfmounts* zeigt an, welche Ressourcen eines oder aller NFS-Server des Netzes bei Clients eingehängt sind.
- Das Kommando *showmount* zeigt an, bei welchen Clients Ressourcen des lokalen oder des angegebenen NFS-Servers eingehängt sind.

4 Kommandos, Dämonen, Verwaltungsdateien

Für die Verwaltung eines verteilten Dateisystems bietet NFS Unterstützung durch:

- Kommandos
- Dämonen
- das Programm *rpcbind*
- Verwaltungsdateien

4.1 NFS-Kommandos

Dieser Abschnitt beschreibt die NFS-Kommandos in alphabetischer Reihenfolge. Die Darstellungsmittel, die in der Kommandosyntax verwendet werden, finden Sie im Abschnitt „Darstellungsmittel“ auf Seite 3.

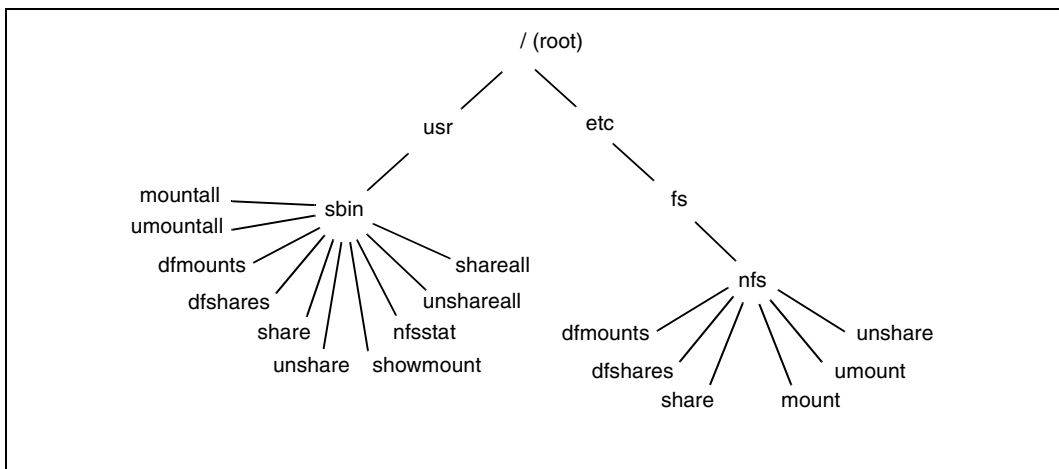


Bild 5: NFS-Kommandos

Die abgebildeten Kommandos benötigen Sie für das Arbeiten mit NFS. Die Kommandos geben Sie wie POSIX-Kommandos in der POSIX-Shell ein.

Die Kommandos *mount* und *umount* sind bereits im Grundausbau der POSIX-Kommandos enthalten, jedoch nur in ihrer generischen Form zur Bearbeitung lokaler Dateisysteme. Mit NFS erhalten sie zusätzliche Optionen und Funktionen.

Die Kommandos *mountall* und *umountall* sind ebenfalls bereits im Grundausbau der POSIX-Kommandos enthalten. Für NFS benötigen sie keine weitere Funktionalität, da sie ihre spezifische Funktionalität einer Eingabedatei entnehmen.

In der folgenden Übersicht sind die NFS-Kommandos aufgelistet:

Kommando	Funktion
dfmounts	Über eingehängte Ressourcen informieren
dfshares	Über bereitgestellte Ressourcen informieren
mount	Ferne Ressourcen einhängen
mountall	Mehrere ferne Ressourcen einhängen
nfsstat	Statistische Informationen ausgeben
share	Lokale Ressourcen für den Client-Zugriff bereitstellen
shareall	Mehrere lokale Ressourcen für den Client-Zugriff bereitstellen
showmount	Über NFS-Clients und Ressourcen informieren
umount	Ferne Ressourcen aushängen
umountall	Mehrere ferne Ressourcen aushängen
unshare	Bereitstellung lokaler Ressourcen zurücknehmen
unshareall	Bereitstellung mehrere lokaler Ressourcen zurücknehmen

Tabelle 3: NFS-Kommandos

dfmounts Über eingehängte Ressourcen informieren

Mit *dfmounts* können Sie feststellen, welche Ressourcen des lokalen Rechners oder eines anderen NFS-Servers auf Client-Systemen eingehängt sind.

Syntax

```
dfmounts[_-F_nfs][_-h][_server][_...]
```

-F_nfs legt fest, dass nur Informationen zu Ressourcen des Dateisystemtyps *nfs* ausgegeben werden. Diese Option müssen Sie nicht angeben, da andere verteilte Dateisysteme zur Zeit nicht unterstützt werden und *nfs* somit der einzige Dateisystemtyp ist, der in der Datei */etc/dfs/dfstypes* aufgeführt ist.

-h unterdrückt die Ausgabe der optionalen Kopfzeile.

server ist der Name eines Rechners, der als NFS-Server Ressourcen bereitstellt; es werden nur Informationen zu den von *server* bereitgestellten Ressourcen ausgegeben. Sie können mehrere *server* angeben.

Wenn *server* nicht angegeben wird, werden Informationen zu den aktuell von Clients eingehängten lokalen Ressourcen ausgegeben.

Ausgabe

Die Ausgabe von *dfmounts* besteht aus einer optionalen Kopfzeile mit Spaltenüberschriften, gefolgt von einer Liste von Zeilen, die die Angaben zu den einzelnen Ressourcen enthalten:

Kopfzeile: RESSOURCE SERVER PATHNAME CLIENTS

RESSOURCE Name der eingehängte Ressource, wie ihn das Kommando *mount* auf dem Client benötigt.

SERVER Name des Systems, das die Ressource bereitstellt.

PATHNAME Pfadname der bereitgestellten Ressource, wie ihn das Kommando *share* auf dem Server benötigt

CLIENTS Namen der Client-Systeme, auf denen derzeit die Ressource eingehängt ist.

Dateien

/etc/dfs/dfstypes Tabelle der installierten Utilities für verteilte Dateisysteme

Beispiel

Sie wollen feststellen, welche Ressourcen, der Rechner *mx4207* bereitstellt, und bei welchen Clients sie eingehängt sind. Geben Sie ein:

► `dfmounts mx4207`

Auf dem Bildschirm erscheint:

RESOURCE	SERVER	PATHNAME	CLIENTS
-	mx4207	/	v214h429
-	mx4207	/home/pope	gabor9,pc4081,pc4083,pc4137,pc4019
-	mx4207	/home/tcp/test	v214h429
-	mx4207	/home/tcp/usr	(anon),v214h429
-	mx4207	/home2/cmx	pc4163,pc4138,pc4186,pc4185, pc4184,pc4076
-	mx4207	/home2/cmx/cmxk10/SRC-WX200/src_CMX	pc4163
-	mx4207	/home2/cmx/dos11_w	pc4163
-	mx4207	/home2/cmx/DOSF70B4	pc4076,pc4163

dfshares Über bereitgestellte Ressourcen informieren

Mit dem Kommando *dfshares* können Sie feststellen, welche Ressourcen auf fernen Systemen für den Client-Zugriff bereitgestellt sind.

Syntax

```
dfshares[_-F_nfs][_h][_server][_...]
```

- F_nfs** legt fest, dass nur Informationen zu Ressourcen des Dateisystemtyps *nfs* ausgegeben werden. Diese Option müssen Sie nicht angeben, da andere verteilte Dateisysteme zur Zeit nicht unterstützt werden und *nfs* somit der einzige Dateisystemtyp ist, der in der Datei */etc/dfs/dfstypes* aufgeführt ist.
- h** unterdrückt die Ausgabe der optionalen Kopfzeile.
- server** ist der Name eines Rechners, der als NFS-Server Ressourcen bereitstellt; es werden nur Informationen zu den von *server* bereitgestellten Ressourcen ausgegeben. Sie können mehrere *server* angeben.
Wenn *server* nicht angegeben wird, werden Informationen zu den Ressourcen ausgegeben, die vom lokalen Rechner für den Client-Zugriff bereitgestellt sind.

Ausgabe

Die Ausgabe von *dfshares* besteht aus einer optionalen Kopfzeile mit Spaltenüberschriften, gefolgt von einer Liste von Zeilen, die die Angaben zu den einzelnen Dateisystemen enthalten:

Kopfzeile: RESSOURCE SERVER ACCESS TRANSPORT

- RESSOURCE** Name der eingehängten Ressource, wie ihn das Kommando *mount* auf dem Client benötigt.
- SERVER** Name des Systems, das die Ressource bereitstellt.
- ACCESS** Zugriffsberechtigungen für die Client-Systeme; da *dfshares* diese Angabe für eine NFS-Ressource nicht ermitteln kann, wird ein Bindestrich (–) ausgegeben.
- TRANSPORT** Transportvorrichtung, über die das Dateisystem gemeinsam benutzt wird; da *dfshares* diese Information für ein NFS-Ressource nicht ermitteln kann, wird ein Bindestrich (–) ausgegeben.

Dateien

/etc/dfs/dfstypes Tabelle der installierten Utilities für verteilte Dateisysteme

Beispiel

Sie wollen feststellen, welche Ressourcen auf dem Rechner *lib* verfügbar sind. Die Informationen sollen mit einer Kopfzeile angezeigt werden. Geben Sie ein:

► `dfshares -F nfs lib`

Auf dem Bildschirm erscheint:

RESOURCE	SERVER	ACCESS	TRANSPORT
lib:/sales	lib	—	—
lib:/usr/mem	lib	—	—
lib:/usr/mod	lib	—	—
lib:/letter	lib	—	—

mount Ferne Ressourcen einhängen

Mit dem Kommando *mount* können Sie sowohl lokale als auch ferne Ressourcen einhängen. Voraussetzung für das Einhängen von fernen Ressourcen ist, dass der ferne NFS-Server die gewünschte Ressource für den Client-Zugriff bereitgestellt hat und erreichbar ist. Die ferne Ressource wird in die lokale Datei-Hierarchie an der Pfadnamenposition *einhängpunkt* eingehängt. *einhängpunkt* muss bereits vorhanden sein. Wenn *einhängpunkt* vor der *mount*-Operation bereits einen Inhalt hat, wird dieser verdeckt, bis *ressource* wieder ausgehängt wird.

Hier ist nur das Einhängen von fernen Ressourcen des Dateisystemtyps *nfs* beschrieben. Wie sie lokale Ressourcen des Dateisystemtyps *ufs* einhängen ist im Handbuch „POSIX-Kommandos“, Kommando *mount* beschrieben.

Wenn die Ressource in der Datei */etc/vfstab* eingetragen ist, genügt es, wenn Sie entweder *ressource* oder *einhängpunkt* angeben. Das Kommando sucht dann in der Datei */etc/vfstab* nach weiteren Angaben.

mount trägt hinzugefügte Dateisysteme in die Tabelle der eingehängten Dateisysteme */etc/mnttab* ein.

Wenn Sie das Kommando ohne Optionen angeben, werden alle aktuell auf ihrem System eingehängten lokalen und fernen Ressourcen aufgelistet.

Berechtigung: Für das Einhängen von fernen Ressourcen benötigen Sie die Root-Berechtigung. Für die Eingabe des Kommandos ohne Optionen benötigen Sie keine besondere Berechtigung.

Syntax

```
mount[_-F_nfs][_r][_o_spez_optionen][_ressource[_einhängpunkt]]
```

- F_nfs** legt fest, dass eine Ressource vom Dateisystemtyp *nfs* eingehängt werden soll. Wenn Sie diese Option nicht angeben, aber *ressource* oder *einhängpunkt* angegeben haben, sucht das Kommando in der Datei */etc/vfstab* nach einem entsprechenden Eintrag und hängt die Ressource mit dem dort angegebenen Dateisystem-Typ ein.
- r** Einhängen der Ressource mit Leseberechtigung.
- o_spez_optionen** ist eine Liste dateisystemspezifischer Optionen, die Sie nach *-o* angeben können. Die einzelnen Optionen der Liste werden durch Kommata getrennt. Sie sind weiter unten beschrieben.

- ressource** gibt die Ressource an, die eingehängt werden soll. Ferne Ressourcen werden in folgender Form angegeben:
server:pfadname
 wobei *server* der Rechnername des NFS-Servers ist, der die Ressource bereitstellt und *pfadname* der absolute Pfadname der Ressource.
- einhängpunkt** Gibt an, wo die *ressource* lokal eingehängt werden soll. Es muss ein absoluter Pfadname angegeben werden.

Folgende *spez_optionen* können Sie nach *-o* angeben:

- rw | ro** *rw* legt Lese- und Schreibzugriff, *ro* legt Nur-Lese-Zugriff auf die eingehängte *ressource* fest. Der Standardwert ist *rw*.
- suid | nosuid** Legt fest, ob gesetzte s-Bits bei der Ausführung beachtet (*suid*) oder ignoriert (*nosuid*) werden. Der Standardwert ist *suid*.
- remount** Bewirkt ein erneutes Einhängen einer bereits eingehängten Ressource, wenn nur Zugriffsberechtigungen geändert wurden.
- bg | fg** Gibt an, ob beim Misslingen des ersten Einhängvorgangs ein erneuter Einhängversuch im Hintergrund (*bg*) oder im Vordergrund (*fg*) ausgeführt werden soll. Der Standardwert ist *fg*.
- retry=*n*** Gibt an, wie oft ein erfolgloser Einhängvorgang wiederholt werden soll. Der Standardwert ist 10 000.
- port=*n*** Gibt die Portnummer des NFS-Servers an. Der Standardwert ist *NFS_PORT*.
- gripid=*GID*** Anlegen einer Datei, deren Gruppennummer (*GID*) der effektiven *GID* des Aufrufers entspricht. Diese Einstellung kann pro Verzeichnis dadurch außer Kraft gesetzt werden, dass das s-Bit für die Gruppe des Vaterverzeichnisses gesetzt wird; in diesem Fall entspricht die Gruppennummer der des Vaterverzeichnisses. Dateien, die in Dateisystemen angelegt werden, die nicht mit der Option *gripid* eingehängt werden, unterliegen der BSD-Semantik, das heißt die *GID* wird unbedingt von der des Vaterverzeichnisses übernommen.
- rsiz=*n*** Definiert die Größe des Lesepuffers in *n* Bytes. Standardwert ist 8 Kbyte.
- wsiz=*n*** Definiert die Größe des Schreibpuffers in *n* Bytes. Standardwert ist 8 Kbyte.
- timeo=*n*** Legt den Wert für die Zeit fest, die der Client maximal auf die Durchführung eines NFS-Auftrags warten soll. *n* wird in Zehntelsekunden angegeben. Standardwert ist 11 Zehntelsekunden.
- retrans=*n*** Setzt die Wiederholungen für einen NFS-Auftrag auf *n* fest. Standardwert ist 5.

- soft | hard** Gibt an, ob bei Nicht-Erreichen des Servers, ein Fehler zurückgegeben werden soll (*soft*) oder ob ein Einhängvorgang solange wiederholt werden soll, bis der Server antwortet (*hard*).
- intr** Legt fest, dass NFS-Aufträge über die Tastatur abgebrochen werden können. Wenn Sie diese Option nicht angeben, ist das Terminal bei einer mit der Option *hard* eingehängten Ressource solange blockiert, bis der Auftrag bearbeitet ist.
- secure** Für NFS-Aufträge wird ein Protokoll mit höherer Sicherheit verwendet (Berechtigungsprüfung `AUTH_DES` von RPC statt Standardberechtigungsprüfung `AUTH_UNIX`). Die Option *secure* muss angegeben werden, wenn der NFS-Server die einzuhängende Ressource mit der Option *secure* bereitgestellt hat, siehe Kommando *share*.
- noac** Unterdrückt die Pufferung von Attributen im Cache-Puffer.
- v2** Hängt Ressourcen von Servern unter Benutzung der NFS-Protokollversion 2 ein, selbst wenn die Server die Protokollversion 3 unterstützen. Diese Option wird in NFS V1.2C nicht unterstützt.

Einhängen im Hintergrund oder Vordergrund

Werden NFS-Dateisysteme mit der Option *bg* eingehängt, bedeutet dies, dass *mount* die Einhängoperation im Hintergrund wiederholen soll, wenn der *mountd*-Dämon des Servers nicht antwortet. *mount* wiederholt die Anforderung so oft, wie in der Option *retry=n* angegeben ist. Sobald das Dateisystem eingehängt ist, warten alle NFS-Anforderungen an den Kernel *timeo=n* Zehntelsekunden auf Antwort. Trifft keine Antwort ein, wird die Wartezeit mit 2 multipliziert, und die Anforderung wird erneut übertragen. Hat die Anzahl der Wiederholungen die in der Option *retrans=n* angegebene Zahl erreicht, gibt ein mit der Option *soft* eingehängtes Dateisystem einen Fehler für die Anforderung zurück; wurde das Dateisystem mit der Option *hard* eingehängt, gibt es eine Warnung aus und wiederholt weiterhin die Anforderung.

Cache-Puffer

Im Cache-Puffer werden Dateiattribute für den Client zwischengespeichert. Attribute zu einer Datei werden nach einer bestimmten Zeit gelöscht. Wird eine Datei geändert, bevor der Cache-Puffer geleert wird, wird das Leerungsintervall um die Zeitspanne seit der letzten Änderung verlängert; dabei wird vorausgesetzt, dass vor kurzem geänderte Dateien bald wieder geändert werden. Für reguläre Dateien und für Verzeichnisse bestehen Mindest- und Höchstwerte für die Verlängerung der Leerungsintervalle.

Dateien

<i>/etc/mnttab</i>	Tabelle der eingehängten Dateisysteme
<i>/etc/dfs/fstypes</i>	Tabelle der installierten Utilities für verteilte Dateisysteme
<i>/etc/vfstab</i>	Tabelle der definierten Dateisysteme

Beispiele

Beispiel 1:

Sie wollen das Dateiverzeichnis */usr/src* des fernen Rechners *serv* auf ihrem lokalen Rechner im Verzeichnis */usr1/proj3/src* einhängen. Auf dem Rechner *serv* wurde das Verzeichnis mit NFS bereitgestellt. Geben Sie ein:

```
▶ mount -F nfs serv:/usr/src /usr1/proj3/src
```

Beispiel 2:

Sie wollen das Verzeichnis *usr/man*, das der Rechner *docgroup* über NFS bereitstellt, auf Ihrem Rechner einhängen. Der Einhängvorgang soll solange wiederholt werden, bis der Server antwortet. Sie wollen das Verzeichnis für Nur-Lese-Zugriff einhängen; Unterbrechung über die Tastatur soll möglich sein. Als Einhängpunkt legen Sie ein Verzeichnis mit demselben Namen *man* an. Geben Sie ein:

```
▶ cd /home1/usr  
  mkdir man  
  mount -F nfs -o ro,hard,intr serv:/usr/man /home1/usr/man
```

mountall Mehrere ferne Ressourcen einhängen

Mit dem Kommando *mountall* können Sie mehrere Ressourcen gleichzeitig einhängen. Die Angaben über die einzuhängenden Ressourcen entnimmt das Kommando einer Eingabedatei, der Datei */etc/vfstab* oder der Standardeingabe. Die Angaben in einer Eingabedatei oder von Standardeingabe müssen dasselbe Format haben wie die Datei */etc/vfstab*.

Wenn Sie keine Option angeben, werden alle Dateisysteme eingehängt, die in der Datei */etc/vfstab* beschrieben sind und bei denen das Feld *automnt* auf *yes* gesetzt ist.

Berechtigung: Das Kommando können Sie nur mit Root-Berechtigung eingeben.

Syntax

mountall[_-F_nfs][_– |_datei]

- F_nfs** legt fest, dass Ressourcen vom Dateisystemtyp *nfs* eingehängt werden sollen. Wenn Sie diese Option nicht angeben, werden alle in der Eingabedatei oder über Standardeingabe spezifizierten Ressourcen eingehängt.
- gibt an, dass das Kommando die Angaben über die einzuhängenden Dateisysteme von der Standardeingabe erwartet. Die einzelnen Zeilen werden mit RETURN bzw. mit EM DUE abgeschlossen. Das Kommando wird nach Eingabe von @ @ d ausgeführt.
- datei Eingabedatei mit Angaben über die einzuhängenden Dateisysteme. Wenn Sie weder – noch *datei* angeben, wird standardmäßig die Datei */etc/vfstab* als Eingabedatei genommen.

Dateien

- /etc/mnttab* Tabelle der eingehängten Dateisysteme
- /etc/vfstab* Tabelle der definierten Dateisysteme

nfsstat Statistische Informationen ausgeben

Das Kommando *nfsstat* gibt statistische Informationen über die NFS-Kommunikation zwischen Client und Server aus. Es werden Informationen ausgegeben über:

- die Anzahl von gesendeten und empfangenen RPCs (Remote Procedure Calls) und den dabei aufgetretenen Fehlern. Es wird unterschieden zwischen den RPCs, die den Rechner als Client und denen, die den Rechner als Server betreffen.
- die Anzahl und Art von NFS-Aufrufen und dabei aufgetretenen Fehlern. Auch hierbei wird unterschieden zwischen denen, die den Rechner als Client und denen die den Rechner als Server betreffen.

Außerdem können Sie mit diesem Kommando die Zähler für die Statistiken reinitialisieren lassen und so den Zeitraum für die Statistiken festlegen.

Wenn Sie das Kommando ohne Optionen eingeben, werden alle statistischen Informationen ausgegeben. Eine Reinitialisierung findet nicht statt.

Syntax

nfsstat[*_-cnrsz*]

- c** Es werden nur Informationen ausgegeben, die den Rechner als Client betreffen.
- n** Es werden nur Informationen über NFS-Aufrufe ausgegeben.
- r** Es werden nur Informationen über RPCs ausgegeben.
- s** Es werden nur Informationen ausgegeben, die den Rechner als Server betreffen.
- z** Veranlasst die Ausgabe und das anschließende Reinitialisieren (auf 0 setzen) der statistischen Informationen.
Die Informationen, die reinitialisiert werden sollen, können mit einem der oben aufgeführten Operanden näher spezifiziert werden.
Wird nur *-z* angegeben, werden alle Informationen reinitialisiert.

RPC-Statistiken: Server

- calls** Anzahl der angekommenen RPCs.
- badcalls** Anzahl der fehlerhaft empfangenen RPCs (die Summe aus *badlen* und *xdr call*).
- nullrecv** Anzahl der nicht verfügbaren, jedoch vermeintlich empfangenen RPCs.
- badlen** Anzahl der empfangenen RPCs mit zu geringer Länge.
- xdr call** Anzahl der empfangenen RPCs, deren Header nicht XDR-entschlüsselt werden konnte.

RPC-Statistiken: Client

calls	Anzahl der RPCs, die abgesetzt wurden.
badcalls	Anzahl der RPCs, die zurückgewiesen wurden.
retrans	Anzahl der innerhalb eines Aufrufs wiederholt gesendeten RPC-Pakete, die erforderlich sind, da keine oder eine fehlerhafte Quittung empfangen wurde.
badxid	Anzahl der Quittungen für RPC-Pakete, die eingetroffen sind, nachdem der RPC schon abgeschlossen war.
timeout	Anzahl der innerhalb eines Aufrufs gesendeten RPC-Pakete, die innerhalb einer bestimmten Zeit nicht beantwortet wurden.
wait	Anzahl der Aufrufe, bei denen auf interne Dateistrukturen gewartet wurde.
newcred	Anzahl der RPCs, für die die Authentisierungsparameter vom Server-Rechner neu angefordert wurden.

NFS-Statistiken

Die NFS-Statistiken sind für Server und Client ähnlich aufgebaut. Es werden Angaben zu den abgesetzten NFS-Aufrufen (*calls*), den fehlgeschlagenen NFS-Aufrufen (*badcalls*) und eine Aufschlüsselung über die Art der abgesetzten NFS-Aufrufe in absoluter Anzahl und Prozentzahl.

Für den Client werden zusätzlich Statistiken über die Anzahl der Anforderung von internen Datenstrukturen (*nclget*) und der hierbei entstandenen Wartesituationen (*nclsleep*).

calls	Anzahl der NFS-Aufträge, die gesendet wurden.
badcalls	Anzahl der fehlgeschlagenen NFS-Aufträge.
nclget	Angabe darüber, wie oft interne Datenstrukturen angefordert wurden.
nclsleep	Angabe darüber, wie oft bei <i>nclget</i> gewartet werden musste.

NFS-Protokollversion 2 und 3

null	keine Aktion (für Testzwecke).
getattr	Attribute für eine Datei anfordern.
setattr	Attribute für eine Datei setzen.
lookup	eine Datei lokalisieren.
readlink	einen symbolischen Verweis lesen.
read	in einer Datei lesen.
wrocache	in den Puffer schreiben.

write	in eine Datei schreiben.
create	eine Datei erzeugen
remove	eine Datei löschen
rename	eine Datei umbenennen
link	einfachen Verweis setzen
symlink	symbolischen Verweis setzen
mkdir	Dateiverzeichnis erzeugen
rmdir	Dateiverzeichnis löschen
readdir	in einem Dateiverzeichnis lesen
fsstat	Dateisysteminformation holen

Nur NFS-Protokollversion 3:

access	Dateizugriffsrechte prüfen
commit	Schreibauftrag stabilisieren
fsinfo	Statische Dateisystem-Information holen
fsstat	Dynamische Dateisystem-Information holen
mknod	Gerätefile erzeugen
pathconf	Informationen zur Pfad-Konfiguration einer Datei holen (maximale Pfadlänge, maximale Anzahl Links usw.)
readdirplus	Erweitertes Lesen eines Dateiverzeichnisses gemäß NFS V3

Beispiele

Beispiel 1:

Sie wollen sich alle statistischen Informationen über NFS und RPC ausgeben lassen, die ihren lokalen Rechner als NFS-Client betreffen. Geben Sie ein:

► `nfsstat -c`

Auf dem Bildschirm erscheint

```
Client rpc:
calls      badcalls   retrans    badxid     timeout    wait       newcred
510690     0          1559       338        1559       0          0
```

Client nfs version 2:

calls	badcalls	nclget	nclsleep		
26	0	26	0		
null	getattr	setattr	root	lookup	readlink
0 0%	12 46%	2 7%	0 0%	8 30%	0 0%
read	wrcache	write	create	remove	rename
0 0%	0 0%	1 3%	1 3%	0 0%	0 0%
link	symlink	mkdir	rmdir	readdir	fsstat
0 0%	0 0%	0 0%	0 0%	1 3%	1 3%

Client nfs version 3:

calls	badcalls	nclget	nclsleep		
510661	0	510654	0		
null	getattr	setattr	lookup	access	readlink
0 0%	152 0%	1 0%	285479 55%	2 0%	0 0%
read	write	create	mkdir	symlink	mknod
59776 11%	60164 11%	7 0%	0 0%	0 0%	0 0%
remove	rmdir	rename	link	readdir	readdirplus
66635 13%	0 0%	0 0%	0 0%	36901 7%	0 0%
fsstat	fsinfo	pathconf	commit		
3 0%	3 0%	2 0%	1537 0%		

Beispiel 2:

Sie wollen sich alle statistischen Informationen über NFS und RPC ausgeben lassen, die ihren lokalen Rechner als NFS-Server betreffen. Geben Sie ein:

► **nfsstat -s**

Server rpc:

calls	badcalls	nullrecv	badlen	xdr call
137514	31720	0	0	0

Server nfs version 2:

calls	badcalls				
1631	0				
null	getattr	setattr	root	lookup	readlink
1 0%	49 3%	0 0%	0 0%	63 3%	3 0%
read	wrcache	write	create	remove	rename
4 0%	0 0%	1479 90%	10 0%	3 0%	0 0%
link	symlink	mkdir	rmdir	readdir	fsstat
0 0%	0 0%	0 0%	0 0%	17 1%	2 0%

Server nfs version 3:

calls	badcalls					
104163	0					
null	getattr	setattr	lookup	access	readlink	
1 0%	287 0%	0 0%	102264 98%	0 0%	2 0%	
read	write	create	mkdir	symlink	mknod	
2 0%	1011 0%	311 0%	2 0%	0 0%	0 0%	
remove	rmdir	rename	link	readdir	readdirplus	
0 0%	0 0%	0 0%	0 0%	278 0%	0 0%	
fsstat	fsinfo	pathconf	commit			
2 0%	2 0%	1 0%	0 0%			

share Lokale Ressourcen für Client-Zugriffe bereitstellen

Mit dem Kommando *share* werden lokale Ressourcen für ferne Client-Zugriffe bereitgestellt.

Wenn Sie das Kommando ohne Option eingeben, werden alle Ressourcen aufgelistet, die aktuell von ihrem System bereitgestellt sind.

Syntax

```
share[_-F_nfs][_o_spez_optionen][_d_beschreibung][_pfadname]
```

-F_nfs	legt fest, dass eine Ressource vom Dateisystemtyp <i>nfs</i> bereitgestellt werden soll. Da in POSIX kein anderer Dateisystem-Typ für verteilte Dateinutzung unterstützt wird, können Sie diese Option weglassen.
-o_spez_optionen	ist eine Liste dateisystemspezifischer Optionen, die Sie nach <i>-o</i> angeben können. Die einzelnen Optionen der Liste werden durch Kommata getrennt. Sie sind weiter unten beschrieben.
-d_beschreibung	ist ein in Anführungszeichen stehender Text, über den Sie Clients Hinweise zur Nutzung der Ressource geben können. Der Text darf keine Sonderzeichen enthalten und nicht länger als 32 Zeichen sein.
pfadname	Pfadname der bereitzustellenden Ressource.

Folgende *spez_optionen* können Sie nach *-o* angeben:

rw	Stellt die Ressource für Lese- und Schreibzugriff bereit. Diese Option ist der Standardwert, d.h. wenn Sie keine <i>spez_optionen</i> angeben, wird allen Clients der Lese-/Schreibzugriff gewährt.
ro	Stellt die Ressource für Nur-Lese-Zugriff bereit.
rw=client[:client]...	Stellt die Ressource den aufgeführten Clients für Lese- und Schreibzugriff bereit. Diese Angabe setzt für einzelne Clients die Unteroption <i>ro</i> außer Kraft.
ro=client[:client]...	Stellt die Ressource den aufgeführten Clients für Nur-Lese-Zugriff bereit. Diese Angabe setzt für einzelne Clients die Unteroption <i>rw</i> außer Kraft.

- anon=*uid*** Weist unbekanntem Benutzern eine User-ID zu. Wenn die Berechtigungsprüfung `AUTH_DES` verwendet wird, wird die angegebene *uid* auf die effektive User-ID (Benutzernummer) nicht berechtigter Benutzer gesetzt. Wenn die Berechtigungsprüfung `AUTH_UNIX` verwendet wird, wird die angegebene *uid* auf Root gesetzt. Standardmäßig wird unbekanntem Benutzern die effektive User-ID `UID_NOBODY` zugewiesen. Wenn *uid* auf `-1` gesetzt ist, wird unbekanntem Benutzern der Zugriff verweigert.
- root=*host*[:*host*]...** gibt an, dass die Root-Benutzer der angegebenen Rechner (*hosts*) am lokalen Rechner ebenfalls die Root-Berechtigung erhalten. Standardmäßig wird keinem Host die Root-Berechtigung gewährt.
- secure** Für NFS-Aufträge wird ein Protokoll mit höherer Sicherheit verwendet (Berechtigungsprüfung `AUTH_DES` von RPC statt Standardberechtigungsprüfung `AUTH_UNIX`). Die Option *secure* muss auch von den Clients beim Einhängen angegeben werden, siehe Kommando *mount*.

Das Kommando wird nicht ausgeführt, wenn für einen Client widersprüchliche Zugriffsrechte definiert sind. Dies ist zum Beispiel der Fall, wenn Sie einen Client-Namen sowohl nach *ro=* als auch nach *rw=* angeben, oder wenn Sie die Optionen *ro* und *rw* zusammen und ohne Argumente angeben.

Dateien

- /etc/dfs/fstypes* Tabelle der installierten Utilities für verteilte Dateisysteme
/etc/dfs/sharetab Tabelle der bereitgestellten Ressourcen

Beispiele

Beispiel 1:

Sie wollen ein Unterverzeichnis von */usr/reports* namens *mtgmemos* über NFS für den Client-Zugriff bereitstellen. Außer NFS ist kein anderes Produkt für verteilten Dateizugriff installiert. Das Verzeichnis soll mit Nur-Lese-Berechtigung für alle Clients zur Verfügung gestellt werden. Geben Sie ein:

```
▶ share -F nfs -o ro -d "MEMOS zu Projekt X" /usr/reports/mtgnotes
```

Beispiel 2:

Sie wollen einen Dateisystem-Ausschnitt als NFS-Ressource für den Client-Zugriff bereitstellen, und zwar aus dem Dateisystem */export* soll das Verzeichnis *graphics* mit allen seinen Unterverzeichnissen und Dateien bereitgestellt werden. Für alle Clients soll Nur-Lese-Zugriff gelten, lediglich der Client „art.dept“ erhält Lese- und Schreibberechtigung. Geben Sie folgendes Kommando ein:

```
▶ share -F nfs -o ro,rw=art.dept /export/graphics
```

shareall Mehrere Ressourcen für den Client-Zugriff bereitstellen

Mit dem Kommando *shareall* können Sie mehrere NFS-Ressourcen gleichzeitig für den Client-Zugriff bereitstellen.

Die Angaben über die bereitzustellenden Ressourcen entnimmt das Kommando einer Eingabedatei, der Datei */etc/dfs/dfstab* oder der Standardeingabe. Die Angaben in einer Eingabedatei oder von Standardeingabe müssen dasselbe Format haben wie die Einträge in der Datei */etc/dfs/dfstab*. Das Format dieser Einträge entspricht der Syntax des Kommandos *share*.

Wenn Sie das Kommando *shareall* ohne Optionen eingeben, werden alle Ressourcen, die in der */etc/dfs/dfstab* eingetragen sind, bereitgestellt.

Syntax

```
shareall[_-F_dstyp [,dstyp...]] [_- | _datei]
```

-F_dstyp [,dstyp...]

legt fest, dass Ressourcen der angegebenen Dateisystem-Typen bereitgestellt werden sollen. Wenn Sie diese Option nicht angeben, werden alle in der Eingabedatei oder über Standardeingabe spezifizierten Ressourcen bereitgestellt.

– gibt an, dass das Kommando die Angaben über die einzuhängenden Dateisysteme von der Standardeingabe erwartet. Die einzelnen Zeilen werden mit RETURN bzw. mit EM DUE abgeschlossen. Das Kommando wird nach Eingabe von @ @ d ausgeführt.

datei Eingabedatei mit Angaben über die bereitzustellenden Ressourcen.

Wenn Sie weder – noch *datei* angeben, wird standardmäßig die Datei */etc/dfs/dfstab* als Eingabedatei verwendet.

Dateien

/etc/dfs/dfstab Tabelle der bereitzustellenden Ressourcen

Beispiele

Beispiel 1:

Sie erstellen eine Eingabedatei namens *misc*, über die drei Ressourcen für Client-Zugriff bereitgestellt werden sollen. Die Datei enthält folgende Einträge:

```
#cat misc
share -F nfs -o ro /usr/reports/mtg.notes
share -F nfs -o ro,rw=art.dept /export/graphics
share -F nfs /usr/man
```

Mit folgendem Kommando werden alle Ressourcen bereitgestellt:

- ▶ `shareall misc`

Beispiel 2:

Sie wollen drei Ressourcen für den Client-Zugriff bereitstellen. Da die Bereitstellung einmalig erfolgt, wird keine Eingabedatei benötigt.

- ▶ `shareall -`

Der Cursor springt in die nächste Zeile. Geben Sie die folgenden Kommandos ein, und schließen Sie jede Kommandozeile mit RETURN bzw. EM DUE ab.

- ▶ `share -F nfs -o ro /usr/reports/mtg.notes`
`share -F nfs -o ro,rw=art.dept /export/graphics`
`share -F nfs /usr/man`

Drücken Sie die Tasten @ @ d. Die Kommandos werden ausgeführt.

showmount Über NFS-Clients und Ressourcen informieren

Mit dem Kommando *showmount* erhalten Sie Informationen über die Namen der Client-Rechner, die Ressourcen des lokalen oder des angegebenen Rechners eingehängt haben. Die Informationen, die *showmount* ausgibt, werden vom Dämon *mountd* am Server-Rechner verwaltet und in der Datei */etc/rmtab* gespeichert.

Wenn Sie keine Optionen angeben, werden die Namen der Client-Rechner ausgegeben, die Dateisysteme eingehängt haben, die vom lokalen Rechner bereitgestellt werden.

Syntax

showmount[_-a][_-d][_-e][_hostname]

- a** Gibt eine Liste im folgenden Format aus:
hostname:directory
wobei *hostname* der Namen des Client-Rechners und *directory* der Namen der eingehängten Ressource ist.
 - d** Gibt die Namen aller Dateisysteme aus, die auf dem Client-Rechner *hostname* eingehängt sind. Die Namen der Client-Rechner werden nicht ausgegeben.
 - e** Es werden die Namen der Dateisysteme ausgegeben, die vom Rechner *hostname* bereitgestellt werden.
- hostname** Name des Rechners, zu dem die Informationen ausgegeben werden sollen. Wenn Sie *hostname* nicht angeben, werden die Informationen zum lokalen Rechner ausgegeben.

Wenn der Systemlauf eines Client-Rechners, der Dateisysteme eines Server-Rechners eingehängt hat, nicht ordnungsgemäß beendet wird, dann wird der Eintrag in der Datei */etc/rmtab* solange aufbewahrt, bis ein Neustart des Systems erfolgt und das Kommando *umount -a* ausgeführt wird.

Dateien

/etc/rmtab Tabelle der eingehängten fernen Ressourcen

Beispiele

Beispiel 1:

Ihr lokaler Rechner heißt *hadern*. Sie wollen feststellen, welche Client-Rechner Ressourcen ihres Rechners eingehängt haben. Geben Sie ein:

► `showmount`

Sie erhalten folgende Ausgabe am Bildschirm:

```
mountainview
tokio
```

Die Client-Rechner *mountainview* und *tokio* haben Ressourcen eingehängt, die der Rechner *hadern* bereitstellt.

Beispiel 2:

Sie möchten wissen, welche Ressourcen ihres lokalen Rechners *hadern* bei Clients eingehängt sind. Geben Sie ein:

► `showmount -d`

Sie erhalten folgende Ausgabe:

```
/usr1/stefan
```

Nur das Dateiverzeichnis */usr1/stefan*, das von ihrem Rechner bereitgestellt wird, ist bei Clients eingehängt.

Beispiel 3:

Sie möchten wissen, welche Client-Rechner welche Ressourcen ihres lokalen Rechners *hadern* eingehängt haben. Geben Sie ein:

► `showmount -a`

Sie erhalten folgende Ausgabe:

```
mountainview:/usr1/stefan
tokio:/usr1/stefan
```

Das Dateiverzeichnis */usr1/stefan*, das von ihrem Rechner bereitgestellt wird, ist sowohl beim Client-Rechner *mountainview* als auch beim Client-Rechner *tokio* eingehängt.

Beispiel 4:

Sie möchten wissen, welche Ressourcen ihres lokalen Rechners *hadern* für den Client-Zugriff mit NFS bereitgestellt sind und mit welchen Zugriffsberechtigungen sie versehen sind. Geben Sie ein:

► `showmount -e`

Sie erhalten folgende Ausgabe:

```
export list for hadern:  
/usr1                (everyone)  
/usr                 grafing
```

Der lokale Rechner *hadern* stellt das Dateiverzeichnis */usr1* für alle Clients (Benutzergruppe *everyone*) und das Dateiverzeichnis */usr* für den Client-Rechner *grafing* bereit.

umount Ferne Ressourcen aushängen

Mit dem Kommando *umount* können Sie lokale und ferne Ressourcen aushängen. Dabei spielt es keine Rolle, ob die Ressource explizit mit dem Kommando *mount* oder implizit über die Datei */etc/vfstab* eingehängt wurde.

Hier ist nur das Aushängen von fernen Ressourcen (Dateisystem-Typ *nfs*) beschrieben. Wie sie lokale Ressourcen (Dateisystem-Typ *ufs*) aushängen ist im Handbuch „POSIX-Kommandos“, Kommando *umount* beschrieben.

Berechtigung: Das Kommando können Sie nur mit Root-Berechtigung eingeben.

Syntax

umount[**-F_nfs**]**_ressource** | **_einhängepunkt**

-F_nfs	legt fest, dass eine Ressource vom Dateisystemtyp <i>nfs</i> ausgehängt werden soll. Wenn Sie diese Option nicht angeben, wird der Dateisystemtyp anhand der Datei <i>/etc/mnttab</i> ermittelt.
ressource	gibt die Ressource an, die ausgehängt werden soll. Ferne Ressourcen vom Typ <i>nfs</i> werden in folgender Form angegeben: <i>server:pfadname</i> wobei <i>server</i> der Rechnername des NFS-Servers, der die Ressource bereitstellt, und <i>pfadname</i> der absolute Pfadname der Ressource ist.
einhängepunkt	Gibt den Einhängpunkt an, an dem die Ressource ausgehängt werden soll. Es muss ein absoluter Pfadname angegeben werden.

Dateien

<i>etc/mnttab</i>	Tabelle der eingehängten Dateisysteme
<i>/etc/dfs/dfstypes</i>	Tabelle der installierten Utilities für verteilte Dateisysteme
<i>etc/vfstab</i>	Tabelle der definierten Dateisysteme

umountall Mehrere ferne Ressourcen aushängen

Mit dem Kommando *umountall* können Sie alle aktuell auf Ihrem System eingehängten Ressourcen aushängen.

Hier ist nur das Aushängen von fernen Ressourcen des Dateisystemtyps *nfs* beschrieben. Wie sie lokale Ressourcen des Dateisystemtyps *ufs* aushängen ist im Handbuch „POSIX-Kommandos“, Kommando *umountall* beschrieben.

Wenn Sie das Kommando ohne Optionen eingeben, werden alle auf Ihrem System eingehängten Ressourcen ausgehängt.

Berechtigung: Das Kommando können Sie nur mit Root-Berechtigung eingeben.

Syntax

umountall[_-F_nfs][_k]

-
- F_nfs** gibt an, dass nur Ressourcen vom Dateisystemtyp *nfs* ausgehängt werden sollen.
 - k** Sendet das Signal SIGKILL an alle Prozesse, die Dateien in der auszuhängenden Ressource geöffnet haben.

Dateien

/etc/mnttab Tabelle der eingehängten Dateisysteme
/etc/vfstab Tabelle der definierten Dateisysteme

unshare **Bereitstellung von Ressourcen zurücknehmen**

Mit dem Kommando *unshareall* können Sie die Bereitstellung von Ressourcen zurücknehmen. Damit werden lokale Ressourcen gegen Einhängerversuche ferner Systeme gesperrt. Für die Ressource muss es in der Datei */etc/dfs/sharetab* einen entsprechenden Eintrag geben.

Syntax

unshare[**-F_nfs**]**_**pfadname

- F_nfs** Gibt an, dass für eine Ressource vom Dateisystemtyp *nfs* die Bereitstellung zurückgenommen werden soll. Wenn Sie diese Option nicht angeben, wird der Dateisystem-Typ aus der ersten Zeile der Datei */etc/dfs/fstypes* entnommen.
- pfadname Gibt den Pfadnamen der Ressource an, die dem Client-Zugriff entzogen werden soll.

Dateien

/etc/dfs/fstypes Tabelle der installierten Utilities für verteilte Dateisysteme
/etc/dfs/sharetab Tabelle der bereitgestellten Ressourcen

Beispiel

Für das automatisch über *dfstab* bereitgestellte Verzeichnis */export/templates* soll die Bereitstellung vorübergehend zurückgenommen werden. NFS-Clients sollen nicht mehr darauf zugreifen können. Geben Sie ein:

► **unshare -F nfs /export/templates**

Mit dem Kommando *share* können Sie das Verzeichnis wieder verfügbar machen.

unshareall Bereitstellung mehrerer Ressourcen zurücknehmen

Mit dem Kommando *unshareall* können Sie die Bereitstellung von mehreren Ressourcen gleichzeitig zurücknehmen. Damit werden lokale Ressourcen gegen Einhängerversuche ferner Systeme gesperrt.

Für die Ressourcen muss es in der Datei */etc/dfs/sharetab* einen entsprechenden Eintrag geben.

Syntax

unshareall[_-F_dstyp[,dstyp...]]

-F_dstyp[dstyp...] Gibt an, dass nur für Ressourcen der angegebenen Dateisystem-Typen die Bereitstellung zurückgenommen werden soll. Wenn Sie diese Option nicht angeben, wird für alle vom lokalen System bereitgestellten Ressourcen die Bereitstellung zurückgenommen.

Dateien

/etc/dfs/dfstab Tabelle der bereitzustellenden Ressourcen

Beispiele

Beispiel 1:

Sie wollen für alle von Ihrem System bereitgestellten NFS-Ressourcen die Bereitstellung zurücknehmen und sie damit für Zugriffe von fernen NFS-Clients sperren. Geben Sie ein:

▶ `unshareall -F nfs`

Beispiel 2:

Sie wollen für alle aktuell von Ihrem System bereitgestellten Ressourcen die Bereitstellung zurücknehmen. Geben Sie ein:

▶ `unshareall`

4.2 Dämonen

Dieser Abschnitt beschreibt die Dämonen in alphabetischer Reihenfolge. Die Darstellungsmittel, die in der Kommandosyntax verwendet werden, finden Sie im Abschnitt „Darstellungsmittel“ auf Seite 3.

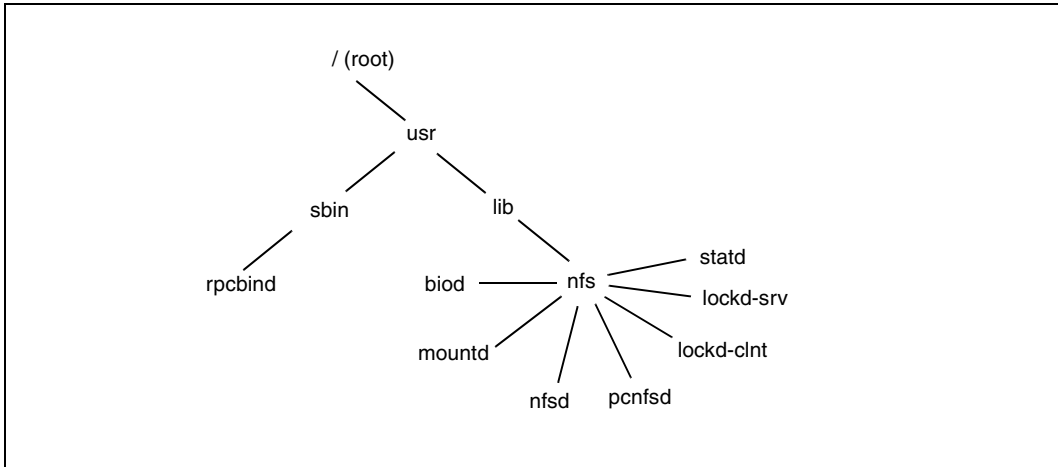


Bild 6: Dämonen

Die NFS-Dämonen *biod*, *mountd*, *nfsd*, *pcnfsd*, *lockd-clnt*, *lockd-srv* und *statd* werden automatisch beim Start von NFS gestartet. Der RPC-Dämon *rpcbind* wird beim POSIX-Start gestartet. Im allgemeinen sollten Sie die Dämonen nur automatisch starten. Wenn Sie die Dämonen jedoch einzeln starten, dann rufen Sie in der POSIX-Shell den Dämon mit dem entsprechenden Pfadnamen auf. Die Dämonen laufen als POSIX-Hintergrundprozesse teils in TU, teils in TPR. Nach ihrem Start „schlafen“ sie, bis sie durch eine Anforderung „geweckt“ werden.

Die Dämonen erledigen die NFS-spezifischen Aufgaben bzw. wickeln die Kommunikation über RPC (Remote Procedure Call) ab. Für die Aufgaben des NFS-Server sind die Dämonen *nfsd* und *mountd* zuständig, für die Aufgaben des NFS-Clients sind die *biod*-Dämonen zuständig. Der *rpcbind*-Dämon ist kein NFS-Dämon, er wird aber zur Abwicklung der Netzkommunikation, die über RPC stattfindet, benötigt und zusammen mit NFS im BS2000 ausgeliefert. Weitere Informationen zu den Aufgaben der Dämonen erhalten Sie in dem Buch „Managing NFS and NIS“.

Standardmäßig laufen auf einem BS2000-System:

```
4 nfsd
4 biod
1 mountd
1 pcnfsd
1 rpcbind
1 statd
1 lockd-srv (auf NLM-Sever und NLM-Client)
1 lockd-clnt (auf NLM-Client)
```

Die Dämonen können Sie überwachen, indem Sie folgendes POSIX-Kommando eingeben:

```
ps -ef | grep dämonname
```

In der folgenden Übersicht sind die einzelnen Dämonen aufgelistet:

Dämon	Funktion
biod	NFS-Client-Dämon für blockorientierte Ein-/Ausgabe
lockd-clnt	Dämon für Network Lock Manager (NLM) Clients
lockd-srv	RPC-Dienst für Network Lock Manager (NLM)
mountd	Dämon zum Einhängen ferner Ressourcen
nfsd	NFS-Server-Dämon für Ein-/Ausgabe
pcnfsd	Dämon zur Unterstützung von DOS-PCs
rpcbind	RPC-Dämon
statd	Status-Monitor für zustandsbehaftete RPC-Dienste

Tabelle 4: Dämonen

biod NFS-Client-Dämon für blockorientierte Ein-/Ausgabe

biod ist ein Dämon für die asynchrone blockorientierte Ein-/Ausgabe. Dieser Dämon bearbeitet Lese- und Schreibaufträge (read-ahead, write-behind) auf einem NFS-Client.

Standardmäßig werden vier *biod*-Dämonen automatisch mit dem Start von NFS gestartet.

Berechtigung: den Dämon *biod* können Sie nur mit Root-Berechtigung starten.

Pfad: /usr/lib/nfs

Syntax

biod[_nservers]

nservers Anzahl der zu startenden Dämonen. Der Standardwert für *nservers* ist vier.

lockd-clnt Dämon für NLM-Clients

lockd-clnt realisiert gemeinsam mit dem Dämon *lockd-srv* den Network Lock Manager (NLM). NLM ist ein Dienst, der die Arbeit mit Dateisperren über NFS unterstützt. *lockd-clnt* muss auf jedem NLM-Clientrechner ablaufen. Er erfüllt dabei die folgenden Aufgaben:

- Wenn der Client seinen ersten NLM-Auftrag absetzt, ermittelt *lockd-clnt* die Adresse des NLM-Servers. Diese Adresse wird vom Client gespeichert, so dass sie nur in Ausnahmefällen (z.B. beim Absturz des Servers) wieder neu durch *lockd-clnt* erfragt werden muss.
- *lockd-clnt* registriert den NLM-Server beim Status-Dämon *statd*.
- *lockd-clnt* zeichnet auf, auf welche NLM-Server der Client zugreift. Die Liste der NLM-Server wird periodisch mit der Einhängetabelle abgeglichen: Wenn eine Eintrag für ein NFS aus der Einhängetabelle gelöscht wird, informiert *lockd-clnt* den Statusdämon *statd* darüber, dass der zugehörige NLM-Server nicht mehr überwacht zu werden braucht.
- *lockd-clnt* etabliert einen RPC-Dienst, der beim Wiederanlauf nach Absturz des Clients durch *statd* aufgerufen wird. Durch diesen Dienst werden die Sperren des Clients beim Server wieder gesetzt.

lockd-clnt wird automatisch beim Start von NFS gestartet. Wenn der Dämon nicht läuft, sollte die Logdatei des Systems auf Fehlermeldungen von *lockd-clnt* überprüft werden.

Pfad: /usr/lib/nfs

Syntax

lockd-clnt

lockd-srv RPC-Dienst für NLM (Network Lock Manager)

lockd-srv realisiert gemeinsam mit dem Dämon *lockd-clnt* den Network Lock Manager (NLM). NLM ist ein Dienst, der die Arbeit mit Dateisperren über NFS unterstützt. *lockd-srv* muss auf jedem NLM-Server und auf jedem NLM-Client laufen. Er bearbeitet NLM-Aufträge ferner Clients. Außerdem registriert *lockd-srv* die Clients beim Statusdämon *statd*.

Eine der Hauptaufgaben von *lockd-srv* ist die Verwaltung blockierter Sperren. Ein Prozess versucht, eine Sperre zu setzen, dies misslingt und der Prozess blockiert. Kann eine Sperre, die vorher blockiert war, auf dem Server gesetzt werden, benachrichtigt *lockd-srv* den *lockd-srv*-Prozess, der auf dem Client läuft. Dieser wiederum sorgt dafür, dass derjenige Client-Prozess fortfahren kann, der blockiert war, weil er die Sperre nicht setzen konnte.

Bei der Initialisierung benachrichtigt *lockd-srv* alle auf dem Server registrierten NLM-Clients mit Hilfe von *statd*. Während einer „Schutzfrist“, die vom Systemverwalter einstellbar ist, können die Clients Sperren wieder setzen, die auf dem Server z.B. aufgrund eines Absturzes verlorengegangen waren. Andere NLM-Aufträge als das Wiedersetzen von Sperren lehnt *lockd-srv* während der Schutzfrist ab.

lockd-srv wird automatisch beim Start von NFS gestartet. Wenn der Dämon nicht läuft, sollte die Logdatei des Systems auf Fehlermeldungen von *lockd-srv* überprüft werden.

Pfad: /usr/lib/nfs

Syntax

lockd-srv[_g schutzfrist][_t.retry_timeout][_f.anzdatei][_s]

- g_schutzfrist** Schutzfrist in Sekunden, während der NLM-Clients verlorengegangene Sperren wieder setzen können. *schutzfrist* hat folgendes Format:
- min:warte:max
- min Minimale Gesamtzeit für die Schutzfrist.
- warte Nach jedem Wiedersetzen einer Sperre erhält *lockd-srv* die Schutzfrist noch mindestens *warte* Sekunden lang aufrecht, solange nicht die maximale Gesamtzeit für die Schutzfrist überschritten wird.
- max Maximale Gesamtzeit für die Schutzfrist.
- Standardwert für die *schutzfrist* ist: 45:10:360
- t.retry_timeout** Wenn Sperren blockieren, versucht *lockd-srv* sie alle *retry_timeout* Sekunden erneut zu setzen. Der Standardwert beträgt 60 Sekunden.

- f** *anzdatei* *anzdatei* bestimmt die Anzahl der Dateien auf dem Server, die von NLM-Clients gesperrt sein können. Der Standardwert ist 1024.
- s** In die Logdatei des Systems wird eine Meldung geschrieben, wenn Konflikte beim Einrichten von „File Shares“ - üblicherweise durch PC-Clients - auftreten.

mountd Dämon zum Einhängen ferner Ressourcen

mountd ist ein Dämon, der auf Anforderungen zum Einhängen von NFS-Ressourcen reagiert. Dazu liest er die Datei */etc/dfs/sharetab*. Dieser Dämon bestimmt, welche Ressourcen auf welchen Rechnern zum Einhängen zu bereitstehen. Er macht auch Angaben darüber, welche Ressourcen von welchen Clients eingehängt wurden. Diese Daten können mit dem Kommando *dfmounts* ausgegeben werden.

Der *mountd*-Dämon wird automatisch mit dem Start von NFS gestartet.

Berechtigung: den Dämon *mountd* können Sie nur mit Root-Berechtigung starten.

Pfad: */usr/lib/nfs*

Syntax

mountd

Dateien

/etc/dfs/sharetab Tabelle der bereitgestellten Ressourcen

nfsd NFS-Server-Dämon für Ein-/Ausgabe

nfsd ist ein Server-Dämon. Er nimmt die Schreib- und Leseanforderungen von Clients entgegen und bearbeitet sie.

Standardmäßig werden vier *nfsd*-Dämonen automatisch beim Start von NFS gestartet.

Berechtigung: den Dämon *nfsd* können Sie nur mit Root-Berechtigung starten.

Pfad: `/usr/lib/nfs`

Syntax

nfsd[`-a`][`-p`protokoll][`-t`gerät][`-anzserver`]

- | | |
|---------------------|---|
| -a | Aktiviert <i>nfsd</i> für alle verfügbaren verbindungslosen Transportprotokolle. |
| -p protokoll | Aktiviert <i>nfsd</i> für das angegebene Protokoll. |
| -t gerät | Aktiviert <i>nfsd</i> für die von dem angegebenen Gerät spezifizierten Transportprotokolle. |
| anzserver | Anzahl der zu startenden Dämonen. <i>anzserver</i> ist entsprechend der Belastung anzugeben, die für diesen NFS-Server erwartet wird. Der Standardwert für <i>anzserver</i> ist vier. |

Dateien

`.nfsXXX` temporäre interne Datei, die vom *nfsd* angelegt wird.

pcnfsd Dämon zur Unterstützung von DOS-PCs

pcnfsd ist ein Dämon zur Unterstützung von ONC-Clients (Open Network Computing) auf PC-Systemen unter dem Betriebssystem Windows9x und Windows NT. Er überprüft die Berechtigungen eines PCs, der sich mit *netopen* beim NFS-Server anmeldet, und unterstützt das Ausdrucken von Dateien des PCs auf einem BS2000-Drucker.

Der *pcnfsd*-Dämon bedient Anforderungen, die über RPC an die Programmnummer 150001 und über das *pcnfsd*-Protokoll V1 gestellt werden.

Der *pcnfsd*-Dämon wird automatisch beim Start von NFS gestartet.

Berechtigung: den Dämon *pcnfsd* können Sie nur mit Root-Berechtigung starten.

Pfad: /usr/lib/nfs

Syntax

pcnfsd

Beispiele

siehe Kapitel „Anbindung eines Windows-PCs“ auf Seite 93

rpcbind Dämon für RPC

rpcbind ist ein Dämon, der die RPC-Programmnummern in allgemeine Adressen umwandelt. Er muss in Betrieb sein, um RPCs absetzen zu können. Da für die NFS-spezifische Netz-Kommunikation RPCs verwendet werden, ist der *rpcbind*-Dämon Voraussetzung für die anderen Dämonen.

Wenn ein Server-Programm, das über RPC kommuniziert (wie z. B. der *mountd*-Dämon), gestartet wird, teilt es dem *rpcbind*-Dämon mit, an welcher Adresse es „horcht“, und welche RPC-Programmnummern es bearbeiten kann. Wenn nun ein Client einen RPC an eine angegebene Programmnummer durchführen will, kontaktiert er zuerst den *rpcbind*-Dämon auf dem Server-Rechner, um die Adresse zu bestimmen, an die er die RPC-Pakete senden soll.

Der *rpcbind*-Dämon wird automatisch beim Start von POSIX über ein RC-Skript gestartet.

Berechtigung: den Dämon *rpcbind* können Sie nur mit Root-Berechtigung starten.

Pfad: `/usr/sbin`

Syntax

rpcbind

statd **Status-Monitor für zustandsbehaftete RPC-Dienste**

statd realisiert den Network Status Monitor (NSM). Er wird vom Lock-Manager zur Wiederherstellung von Sperr-Tabellen nach Rechnerabstürzen benötigt und muss auf dem Client und auf dem Server laufen. *statd* arbeitet zur Bereitstellung von Wiederanlauffunktionen nach Systemabstürzen mit zustandsbehafteten RPC-Dämonen zusammen.

Pfad: /usr/lib/nfs

Syntax

statd

Dateien

<i>/etc/sm</i>	enthält Namen der Clients, die eine oder mehrere Dateien gesperrt haben.
<i>/etc/sm.bak</i>	enthält Namen der Clients, die von einem Neustart des Servers benachrichtigt werden müssen.

4.3 rpcinfo-Programm

Der RPC-Mechanismus (Remote Procedure Call) basiert auf dem Client/Server-Modell. Ein Server bietet Dienste an und gibt dem RPC-Dämon bekannt, an welcher Adresse er auf RPCs von Clients wartet und über welche Protokolle mit ihm kommuniziert werden kann. Der Client setzt einen RPC ab, um so übers Netz den Dienst des Servers zu nutzen. Er benutzt den RPC-Dämon, um den Kontakt mit dem Server herzustellen. Die folgenden vier Werte spielen hierbei eine Rolle:

- Programmnummer
- Versionsnummer
- Prozedurnummer
- Protokoll

Dieser Abschnitt enthält eine Beschreibung des *rpcinfo*-Programms. Die Darstellungsmittel, die in der Kommandosyntax verwendet werden, finden Sie im Abschnitt „Darstellungsmittel“ auf Seite 3.

rpcinfo Ausgeben von RPC-Informationen

rpcinfo führt einen RPC an einen *rpcbind*-Dämon (portmapper) aus und listet das Ergebnis auf.

Wenn Sie das Kommando *rpcinfo* mit der Option *-p* angeben, werden alle RPC-Dienste aufgelistet, die beim *rpcbind*-Dämon registriert sind.

Wenn Sie das Kommando *rpcinfo* mit der Option *-T* angeben, führt *rpcinfo* einen RPC an die Prozedur 0 von *programm* und *version* auf dem angegebenen *host* aus und meldet, ob eine Antwort empfangen wurde. *transport* ist der Transportweg, der zur Kontaktierung des angegebenen Dienstes benutzt werden muss. Die ferne Adresse des Dienstes erhält man durch einen Aufruf an den fernen *rpcbind*-Dämon.

Pfad: /usr/bin

Syntax

```

rpcinfo[_host]
rpcinfo_-p[_host]
rpcinfo_-T_transport_host_programm_version
rpcinfo[_-n_portnum]_-u_host_programm_version
rpcinfo[_-n_portnum]_-t_host_programm_version
rpcinfo_-a_serv_adresse_-T_transport_programm[_version]
rpcinfo_-b[_-T_transport]_programm_version
rpcinfo_-d[_-T_transport]_programm_version
  
```

- | | |
|--------------|--|
| host | Name eines fernen Rechners. Standardwert ist der lokale Rechner. Wenn Sie das Kommando <i>rpcinfo</i> mit <i>host</i> angeben, werden alle registrierten RPC-Dienste mit <i>rpcbind</i> und <i>host</i> aufgelistet. |
| -T_transport | Gibt den Transportweg an, auf dem die Auskunft gefordert wird. Ist diese Option nicht angegeben, benutzt <i>rpcinfo</i> den Transportweg, der in der Umgebungsvariablen <i>NEPATH</i> , oder, wenn diese nicht gesetzt oder Null ist, in der Netzkonfigurationsdatenbank angegeben ist. Es handelt sich hierbei um eine generische Option, die in Verbindung mit jeder anderen Option verwendet werden kann, mit Ausnahme der Option <i>-b</i> . |

- a**_{serv_adresse} Verwendet *serv_adresse* als die (allgemeine) Adresse für den Dienst auf *transport*, um mit dem Kommando *ping* eine Statusabfrage der Prozedur 0 des angegebenen *programms* durchzuführen und anzuzeigen, ob eine Antwort empfangen wurde. Die Verwendung der Option *-a* ist an die Verwendung der Option *-T* gebunden.
Ist die Versionsnummer nicht angegeben, versucht *rpcinfo*, mit dem Kommando *ping* alle verfügbaren Versionsnummern für diese Programmnummer herauszufinden. Diese Option vermeidet Aufrufe an *rpcbind* auf fernen Rechnern zum Auffinden der Adresse des Dienstes. Die *serv_adresse* hat das Format der allgemeinen Adresse des angegebenen Transportweges.
- b** Führt einen RPC-Broadcast an die Prozedur 0 des angegebenen *programms* und der angegebenen *version* durch und meldet alle antwortenden Hosts. Ist *transport* angegeben, übermittelt *rpcinfo* die Anforderung nur auf dem durch *transport* spezifizierten Transportweg.
- d** Löscht die Registrierung für den RPC-Dienst des angegebenen *programms* und der angegebenen *version*. Ist *transport* angegeben, so wird die Registrierung nur auf diesem Transportweg gelöscht, ansonsten auf allen Transportwegen, auf denen der RPC-Dienst registriert war. Diese Option kann nur von einem privilegierten Benutzer verwendet werden.
- n**_{portnum} Verwendet *portnum* als die Portnummer für die Optionen *-t* und *-u* anstatt der durch den *rpcbind* angegebenen Portnummer. Durch die Verwendung dieser Option wird ein Aufruf an den fernen *rpcbind* zum Herausfinden der Adresse des Dienstes vermieden.
- p** Prüft den *rpcbind*-Dämon auf *host* und gibt eine Liste aller registrierten RPC-Programme aus. Ist *host* nicht angegeben, ist der lokale Host der Standardhost.
- t** Führt unter Verwendung von TCP einen RPC-Aufruf an die Prozedur 0 von *programm* auf dem angegebenen *host* durch, und meldet, ob eine Antwort empfangen wurde.
- u** Führt unter Verwendung von UDP einen RPC-Aufruf an die Prozedur 0 von *programm* auf dem angegebenen *host* durch, und meldet, ob eine Antwort empfangen wurde.
- programm** Programmnummer, angegeben als Zahl.

version Ist eine *version* angegeben, versucht *rpcinfo*, diese *version* des angegebenen *programms* aufzurufen. Ansonsten versucht *rpcinfo*, alle registrierten Versionsnummern für das angegebene *programm* zu finden, indem die als nicht existent angenommene Version 0 aufgerufen wird; existiert sie jedoch, versucht *rpcinfo*, diese Information durch den Aufruf einer extrem hohen Versionsnummer zu erhalten und jede registrierte Version aufzurufen. Bitte beachten Sie, dass die Versionsnummer für die Optionen *-b* und *-d* gefordert wird.

Beispiele

Beispiel 1:

Zur Anzeige aller auf dem lokalen Rechner registrierten RPC-Dienste verwenden Sie:

► `rpcinfo`

Sie erhalten folgende Ausgabe:

program	version	netid	address	service	owner
100000	3	udp	0.0.0.0.0.111	portmapper	superuser
100000	2	udp	0.0.0.0.0.111	portmapper	superuser
100000	3	ticotsord	BS2TEST1.rpc	portmapper	superuser
100000	3	ticots	BS2TEST1.rpc	portmapper	superuser
100000	3	ticlts	BS2TEST1.rpc	portmapper	superuser
100003	2	ticlts	BS2TEST1.nfsd	nfs	superuser
100003	2	udp	0.0.0.0.8.1	nfs	superuser
100005	1	ticlts	\007\000\000\000	mountd	superuser
100005	1	udp	0.0.0.0.16.0	mountd	superuser
100005	1	ticots	\003\000\000\000	mountd	superuser
100005	1	ticotsord	\003\000\000\000	mountd	superuser
100005	1	tcp	0.0.0.0.16.1	mountd	superuser
150001	1	udp	0.0.0.0.2.127	pcnfsd	superuser

Beispiel 2:

Zur Anzeige aller mit *rpcbind* auf dem Rechner namens *klaxon* registrierten RPC-Dienste verwenden Sie:

► `rpcinfo klaxon`

Beispiel 3:

Um zu zeigen, ob der RPC-Dienst mit der Programmnummer *prog_no* und der Version *vers* auf dem Rechner namens *klaxon* für den Transportweg *tcp* registriert ist, verwenden Sie:

► `rpcinfo -T tcp klaxon prog_no vers`

Beispiel 4:

Zur Anzeige aller beim Anschlussabbilder auf dem lokalen Rechner registrierten RPC-Dienste verwenden Sie:

► `rpcinfo -p`

Sie erhalten folgende Ausgabe:

```
program vers proto port
100000 3 udp 111 portmapper
100000 2 udp 111 portmapper
100003 2 udp 2049 nfs
100005 1 udp 4096 mountd
100005 1 tcp 4097 mountd
150001 1 udp 639 pcnfsd
```

Beispiel 5:

Um den Status der Version 2 von *rpcbind* (Programmnummer 100000) mit dem Kommando *ping* auf dem Host *sparky* abzufragen:

► `rpcinfo -t sparky 100000 2`

Beispiel 6:

Zum Löschen der Registrierung für die Version 1 des *walld*-Dienstes (Programmnummer 100008) für alle Transportwege benutzen Sie:

► `rpcinfo -d 100008 1`

4.4 Verwaltungsdateien

Dieser Abschnitt beschreibt die Verwaltungsdateien.

Die Darstellungsmittel, die in der Kommandosyntax verwendet werden, finden Sie im Abschnitt „Darstellungsmittel“ auf Seite 3.

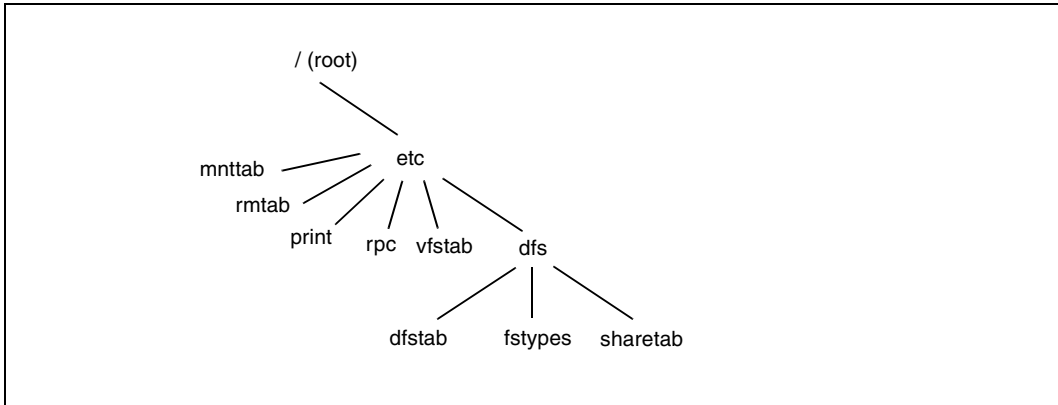


Bild 7: Verwaltungsdateien

Für den Betrieb von NFS werden einige Verwaltungsdateien benutzt. Sie werden entweder von NFS oder von POSIX beim Start automatisch angelegt. Die Dateien werden für folgende Verwaltungsaufgaben benutzt:

- für die automatische Verwaltung von Ressourcen werden die Dateien `/etc/print`, `/etc/rpc`, `/etc/vfstab`, `/etc/dfs/dfstab` und `/etc/dfs/fstypes` benutzt. In diese Dateien tragen Sie Standardwerte ein, die von den Kommandos zur NFS-Ressourcenverwaltung interpretiert werden.
- für Information über Ressourcen werden die Dateien `/etc/dfs/sharetab`, `/etc/mnttab` und `/etc/rmtab` benutzt. In diese Dateien protokollieren die Kommandos, die für die Verwaltung von NFS-Ressourcen eingegeben wurden, ihre Aktionen.

Die Dateien können Sie nur mit Root-Berechtigung verändern. Sie rufen hierzu in der POSIX-Shell den Editor `edt` auf. Siehe hierzu das Handbuch „POSIX-Kommandos“.

In der folgenden Übersicht sind die Verwaltungsdateien aufgelistet:

Pfad	Datei	Funktion
/etc	mnttab	Tabelle der eingehängten Dateisysteme
/etc	print	Schablonen für BS2000-Druckkommandos
/etc	rmtab	Tabelle der eingehängten fernen Ressourcen
/etc	rpc	RPC-Programmnummern-Datei
/etc	vfstab	Tabelle der definierten Dateisysteme
/etc/dfs	dfstab	Tabelle der bereitzustellenden Ressourcen
/etc/dfs	fstypes	Tabelle der installierten Utilities für verteilte Dateisysteme
/etc/dfs	sharetab	Tabelle der bereitgestellten Ressourcen

Tabelle 5: Verwaltungsdateien

/etc/mnttab Tabelle der eingehängten Dateisysteme

Die Datei */etc/mnttab* enthält Angaben über alle am lokalen Rechner eingehängten Dateisysteme. Diese Datei enthält Informationen, die durch das Kommando *mount* erzeugt werden.

Jede Zeile enthält folgende Informationen, die durch eine beliebige Anzahl von Leerzeichen und/oder Tabulatoren getrennt sind:

Aufbau

ressource	mountp	fstyp	spec-options	time
ressource	absoluter Pfadname des eingehängten Dateisystems. Ferne Ressourcen haben die Form: <i>server:pfadname</i> wobei <i>server</i> der Rechnername des NFS-Servers ist, der die Ressource bereitstellt und <i>pfadname</i> der absolute Pfadname der Ressource.			
mountp	absoluter Pfadname des Einhängepunkts.			
fstype	Dateisystem-Typ.			
spec-options	Optionen, wie sie beim <i>mount</i> -Kommando angegeben wurden.			
time	Einhängezeitpunkt, angegeben in Sekunden seit 1.1.1970			

Beispiel

Geben Sie in der POSIX-Shell ein: `cat /etc/mnttab`

```
/dev/root      /           ufs  rw,suid      802532552
/dev/proc      /proc      proc rw,          802532553
/dev/fd        /dev/fd    fdfs rw           802532553
/dev/dsk/3     /var       ufs  suid,rw,noquota 802532558
/dev/dsk/2     /home1     ufs  suid,rw,noquota 802532588
SINTEST1:/nfs /nfsclient ufs  rw           802536261
```

/etc/print Schablonen für BS2000-Druckkommandos

Die Datei */etc/print* enthält Schablonen für BS2000-Druckkommandos (siehe hierzu das Kommando PRINT-FILE im Handbuch „BS2000/OSD-BC, Kommandos“). Jede Zeile der Datei besteht aus den folgenden Feldern:

Aufbau

```
name:code:pname:text:device:form:space:laser:chars:image:xstring:
```

- name** Name für den BS2000-Drucker, wie er beim *prinit*-Kommando im DOS benutzt wird. Dieser Name kann maximal 8 Buchstaben lang sein.
- code** Legt fest, ob und wie die Druckdatei nach EBCDIC konvertiert werden soll. Hierzu geben sie an, welche Konvertierungstabelle benutzt werden soll. Die Konvertierungstabellen befinden sich im Verzeichnis */usr/lib/conv*. Siehe hierzu auch das Handbuch „POSIX-Kommandos“, Kommando *iconv*.

code	Tabelle	Auswirkung
keine Angabe		Standardverhalten (wie 646)
n		es wird nicht konvertiert
646	646.edf03.t	konvertiert von ASCII-646 nach EBCDIC-edf03; Standardverhalten
646da	646da.8859.t	konvertiert von ASCII-646da nach EBCDIC-edf03;
646fr	646fr.8859.t	konvertiert von ASCII-646fr nach EBCDIC-edf03;
646de	646de.8859.t	konvertiert von ASCII-646de nach EBCDIC-edf03;
646it	646it.8859.t	konvertiert von ASCII-646it nach EBCDIC-edf03;
646en	646en.8859.t	konvertiert von ASCII-646en nach EBCDIC-edf03;
646sv	646sv.8859.t	konvertiert von ASCII-646sv nach EBCDIC-edf03;
646es	646es.8859.t	konvertiert von ASCII-646es nach EBCDIC-edf03;
8859	8859.edf04.t	konvertiert von ASCII-8859-1 nach EBCDIC-edf04;

Tabelle 6: Konvertierungstabellen

- pname** Name des SPOOLOUT-Jobs im BS2000. Der Standardwert ist PCNFS.
- text** Text für die Headerpage
Dieser Name kann bis zu 32 Buchstaben haben (D'text'). Es gibt keinen Standardwert.
- device** Gerätenamen für den BS2000-Drucker. Es gibt keinen Standardwert.

form	Formcode für Laser-Drucker. Es gibt keinen Standardwert.
space	Steuerung des Zeilenvorschubs für den Drucker. Gültige Werte: 1,2,3,E,A,I 1,2,3 Zahl der Zeilenvorschübe nach jeder Druckzeile. E Die Datei enthält Siemens Nixdorf-konforme Zeilenvorschub-Zeichen. A Die Datei enthält ASA-konforme Zeilenvorschub-Zeichen. I Die Datei enthält Industriestandard-konforme Zeilenvorschub-Zeichen. Es gibt keinen Standardwert.
laser	Die Datei wird über einen Laserdrucker ausgegeben (<i>chars</i> und <i>image</i> werden ausgewertet). Der Standardwert beträgt <i>YES</i> .
chars	Bezeichnet einen oder mehrere Zeichensätze, die zum Drucken der Datei verwendet werden sollen.
image	Operand für die Ausgabe auf Laserdrucker. Bezeichnet eine Benutzerdatei, die LOOP-, Zeichen- und POOL-Sätze enthalten kann. Fehlt der Operand, so werden die entsprechenden Parameter der Datei <i>\$TSOS.NDFILE</i> , <i>\$TSOS.HPFILE</i> bzw. <i>\$TSOS.RSOFIL</i> e entnommen.
xstring	Erweiterte Angaben für das BS2000-PRINT-Kommando. Zeichenkette mit maximal 224 Zeichen. Es gibt keinen Standardwert.

Beispiele

siehe Kapitel „Anbindung eines Windows-PCs“ auf Seite 93.

/etc/rmtab Tabelle der eingehängten fernen Ressourcen

Die Datei */etc/rmtab* enthält Angaben über alle am lokalen Rechner eingehängten fernen Ressourcen. Diese Datei enthält Informationen, die durch das Kommando *mount -F nfs* erzeugt wurden.

Jede Zeile enthält folgende Informationen:

Aufbau

ressource

ressource absoluter Pfadname des eingehängten Dateisystems vom Typ *nfs*. Die Angabe hat die Form:
server:pfadname
wobei *server* der Rechnername des NFS-Servers ist, der die Ressource bereitstellt und *pfadname* der absolute Pfadname der Ressource.

Beispiel

Geben Sie in der POSIX-Shell ein: `cat /etc/rmtab`

```
SINTEST1:/nfs1/nfsserver
```

/etc/rpc RPC-Programmnummern-Datei

Die RPC-Programmnummer-Datei enthält von Benutzern lesbare Namen, die anstelle von RPC-Programmnummern intern verwendet werden können.

Jede Zeile enthält folgende Informationen, die durch eine beliebige Anzahl von Leerzeichen und/oder Tabulatoren getrennt sind:

Aufbau

```
programmname    programmnummer    aliases
#  text
```

programmname Name des RPC-Server-Programms

programmnummer RPC-Programmnummer

aliases Aliasnamen, die anstelle der Programmnummer verwendet werden können

text Kommentar: ab dem Sonderzeichen # bis zum Zeilenende wird *text* als Kommentar interpretiert.

Beispiel

Geben Sie in der POSIX-Shell ein: `cat /etc/rpc`

```
event          100101  na.event      # SunNet Manager
async-llckmgr  100020
logger         100102  na.logger     # SunNet Manager
async-nlockmgr 100021
x25.inr        100022
sync           100104  na.sync
statmon        100023
status         100024

.
.
.

ping           100115  na.ping
rpcnfs         100116  na.rpcnfs
hostif         100117  na.hostif
tfsd           100037
portmapper     100000  portmap sunrpc
etherif        100118  na.etherif
nsed           100038
rstatd         100001  rstat rup perfmeter
nsemntd        100039
rusersd        100002  rusers
nfs            100003  nfsprog
ypserv         100004  ypprog
mountd         100005  mount showmount
sync_nlockmgr  200004
pcnfsd         150001  pcnfs
sync_llckmgr   200005
ypbind         100007
walld          100008  rwall shutdown
yppasswdd      100009  yppasswd
ioadmd         100055  rpc.ioadmd
```

/etc/vfstab Tabelle der definierten Dateisysteme

Die Datei */etc/vfstab* beschreibt Standardwerte für jedes auf dem lokalen Rechner definierte Dateisystem. Die Datei können Sie mit *edt* bearbeiten.

Die Dateisysteme, die in der Datei */etc/vfstab* eingetragen sind, werden beim Start von POSIX eingehängt oder wenn Sie das Kommando *mountall* ohne die Option *-* oder *datei* eingeben. Dateisysteme vom Typ *nfs* hängen Sie nach dem Start von NFS mit *mountall -F nfs* ein.

Die Felder in der Tabelle sind durch Leerzeichen getrennt. Ein Bindestrich (*-*) kennzeichnet einen leeren Eintrag im Feld. Die Tabelle enthält folgende Felder:

Aufbau

```
special_fsckdev_mountp_fstype_fsckpass_automnt_mntopts
```

special	beschreibt die einzuhängende Ressource. Für Dateisysteme vom Typ <i>nfs</i> hat <i>special</i> folgende Form: <i>server:pfadname</i> wobei <i>server</i> der Rechnername des NFS-Servers ist, der die Ressource bereitstellt und <i>pfadname</i> der absolute Pfadname der Ressource.
fsckdev	Name des blockorientierten Geräts bzw. der Ressource des zeichenorientierten Geräts. Beim Einhängen von fernen Ressourcen wird dieser Parameter nicht unterstützt und muss durch den Bindestrich (<i>-</i>) ersetzt werden.
mountp	Einhängepunkt: absoluter Pfadname des Verzeichnisses, in dem die Ressource eingehängt werden soll.
fstype	Dateisystem-Typ: für ferne Ressourcen tragen Sie <i>nfs</i> ein.
fsckpass	ist die für mehrere <i>fsck</i> -Kommandos zu verwendende Durchlaufnummer. Beim Einhängen von fernen Ressourcen wird dieser Parameter nicht unterstützt und muss durch den Bindestrich (<i>-</i>) ersetzt werden.
automnt	gibt an, ob die Ressource durch <i>mountall</i> automatisch beim Start von POSIX eingehängt werden soll (<i>yes</i>) oder nicht (<i>no</i>). Dateisysteme vom Typ <i>nfs</i> werden beim Start von POSIX nicht eingehängt. Verwenden Sie zum Einhängen von <i>nfs</i> -Dateisystemen nach dem Start von NFS das Kommando <i>mountall -F nfs</i> .
mntopt	Liste durch Kommata getrennter Optionen für das Einhängen des Dateisystems. Die Optionen entsprechen den <i>spez_optionen</i> des Kommandos <i>mount</i> .

Beispiel

Geben Sie in der POSIX-Shell ein: `cat /etc/vfstab`

```

/dev/root      /dev/rroot    /                ufs      1        yes      -
/proc -       /proc proc     -         no       -
/dev/fd -     /dev/fd fdfs    -         no       -
/dev/dsk/3    /dev/rdsk/3   /var         ufs      1        yes      -
/dev/dsk/2    /dev/rdsk/2   /kml        ufs      1        yes      -
/dev/dsk/4    /dev/rdsk/4   /nfs1       ufs      1        yes      -
/dev/dsk/5    /dev/rdsk/5   /tmp1       ufs      1        yes      -
/dev/dsk/6    /dev/rdsk/6   /vsx        ufs      1        no       -
/dev/dsk/7    /dev/rdsk/7   /vsx-ro     ufs      1        no       -
/dev/dsk/8    /dev/rdsk/8   /usr1       ufs      1        yes      -
/dev/dsk/9    /dev/rdsk/9   /usr2       ufs      1        yes      -
/dev/dsk/10   /dev/rdsk/10  /home1      ufs      1        no       -
/dev/dsk/11   /dev/rdsk/11  /home2      ufs      1        no       -
/dev/dsk/12   /dev/rdsk/12  /home3      ufs      1        no       -
/dev/dsk/13   /dev/rdsk/13  /home4      ufs      1        no       -
/dev/dsk/14   /dev/rdsk/14  /home5      ufs      1        no       -
/dev/dsk/15   /dev/rdsk/15  /home6      ufs      1        no       -
/dev/dsk/16   /dev/rdsk/16  /oldroot    ufs      1        no       -
/dev/dsk/17   /dev/rdsk/17  /ascii     ufs      1        yes      -
tanz:/usr/local -      /usr/local/tmp nfs      -        yes      ro

```

/etc/dfs/dfstab **Tabelle der bereitzustellenden Ressourcen**

Die Datei */etc/dfs/dfstab* enthält *share*-Kommandos zum Bereitstellen von Dateisystemen. Die *share*-Kommandos, die in der Datei */etc/dfs/dfstab* eingetragen sind, werden ausgeführt, wenn Sie das Kommando *shareall* ohne Angabe einer Eingabedatei eingeben.

Aufbau

```
share[_F_nfs][_o_spez_optionen][_d_beschreibung][_p_fadname]
```

Der Aufbau der Zeilen dieser Datei entspricht der Syntax des Kommandos *share*, siehe Seite 53.

/etc/dfs/fstypes

Tabelle der installierten Utilities für verteilte Dateisysteme

Die Datei */etc/dfs/fstypes* enthält eine Liste der am System installierten Utility-Pakete für verteilte Dateisysteme.

Wenn NFS-Kommandos ohne die Option *-F nfs* gestartet werden, verwendet das System als Standardwert den in der ersten Zeile dieser Datei aufgeführten Dateisystem-Typ. Die Datei kann mit *edt* editiert werden.

Aufbau

fstyp bezeichnung: version

fstyp	Dateisystemtyp
bezeichnung	Name des Utilitypaketes
version	Version des Utilitypaketes

Beispiel

Wenn NFS V3.0 das einzige am System installierte Utility-Paket für verteilte Dateisysteme ist, enthält diese Datei als erste und einzige Zeile folgendes:

```
nfs Network File System Utilities: 3.0
```

/etc/dfs/sharetab Tabelle der bereitgestellten Ressourcen

Die Datei */etc/dfs/sharetab* enthält eine Tabelle der lokalen Ressourcen, die durch den Befehl *share* für den Client-Zugriff bereitgestellt wurden.

Jede Zeile der Tabelle enthält folgende Felder; die Felder in der Tabelle sind durch Leerzeichen getrennt; ein Bindestrich (–) kennzeichnet einen leeren Eintrag:

Aufbau

pfadname_ressource_fstyp_spez_opt_beschreibung

pfadname Pfadname der bereitgestellten Ressource.

ressource Symbolischer Namen, über den ferne Rechner auf die Ressource zugreifen können.

fstyp Dateisystem-Typ der bereitgestellten Ressource.

spez_optionen Die dateisystemspezifischen Optionen, die bei der Bereitstellung angegeben wurden.

beschreibung Kommentar, der die bereitgestellte Ressource beschreibt.

Beispiel

Geben Sie in der POSIX-Shell ein: `cat /etc/sharetab`

```
/nfs1/nfsserver -      nfs      rw
```

5 Anbindung eines Windows-PCs

Das Produkt NFS V3.0 ermöglicht für Intel-basierte Personal Computer (PCs), die mit einem Windows Betriebssystem betrieben werden, den Zugriff auf das POSIX-Dateisystem eines BS2000-Rechners und das Ausdrucken von Dateien auf einem an den BS2000-Rechner angeschlossenen Drucker.

Im Unterschied zu den UNIX-Implementierungen für Intel-PCs (z.B. SCO-UNIX, Solaris, Linux) ist NFS in den Windows Betriebssystemen (Windows 95, Windows 98, Windows NT und Windows 2000) nicht integriert.

Um einen Windows-PC als NFS-Server oder Client einzusetzen, muss ein entsprechendes Zusatzprodukt installiert werden. Typischerweise wird ein PC als NFS-Client betrieben, der freigegebene Server-Ressourcen im POSIX-Dateisystem als Netzlaufwerk einhängt.

Neben Implementierungen von anderen Herstellern, die NFS-Dienste für Windows-PCs enthalten, steht von Fujitsu Siemens das Produkt LAN1 Pro zur Verfügung.

LAN1 Pro basiert auf LAN WorkPlace Pro von Novell und besteht aus einer Suite von TCP/IP Applikationen (Internet Browser, Terminal-Emulationen, Filetransfer (FTP), NFS-Client, X-Server und Email). Der NFS-Client von LAN1 Pro ermöglicht einen transparenten Zugriff auf freigegebene POSIX-Verzeichnisse und auf Drucker in BS2000/OSD.

Im Folgenden wird der Einsatz des in LAN1 Pro enthaltenen NFS-Clients beschrieben.

5.1 Vorbereitungen

Vor dem Einsatz des NFS-Clients auf dem PC müssen Sie im BS2000 und auf dem PC einige Vorbereitungen treffen.

Im BS2000

Für die POSIX-Kennung, über die der PC die Verbindung zum POSIX herstellt, müssen Sie mit dem Kommando */MODIFY-USER-ATTRIBUTES* eine Standard-Abrechnungsnummer für *rlogin* zuordnen.

Auf dem PC

Während der Installation von LAN1 Pro über die Autorun-Applikation der ausgelieferten CD müssen sie eine manuelle Auswahl der zu installierenden Dienste wählen ("Custom"-Installation). Bei der Installationsvariante "Typical" wird NFS-Client nicht installiert.

Nach Abschluss der eigentlichen Installation werden Sie zur Konfiguration des NFS-Clients aufgefordert. Hierbei ist folgendes zu beachten:

- Im angebotenen Dialog-Fenster "Default User-Information" können Sie Angaben über die standardmäßig beim Einhängen von NFS-Dateisystemen zu verwendende Zugangsberechtigung machen.

Als Benutzername muss die BS2000-Kennung angegeben werden, über die der PC die Verbindung zum POSIX herstellt.

Das zur BS2000-Kennung gehörige Passwort muss in Großbuchstaben angegeben werden. BS2000 akzeptiert nur Großbuchstaben in Passwörtern, eine automatische Konvertierung ist nicht gegeben.

Eine Anmeldung ohne Passwort wird vom *pcnfs*-Dämon standardmäßig abgewiesen. Soll jedoch mit einer Kennung ohne Passwort gearbeitet werden, so kann, wenn SECOS installiert ist, die Passwortüberprüfung mit folgendem Kommando ausgeschaltete werden:

```
/MODIFY_LOGON_PROTECTION ,POSIX_RLOGIN_ACCESS=*YES -  
/ (PASSWORD-CHECK=*NO)
```

- Die Angabe von NFS-Servern im Dialogschirm "Servers List" ist optional. Fall Sie hier Namen von Server-Hosts angeben, werden diese Server und deren freigegebene Verzeichnisse auf dem Dialogschirm "Netzlaufwerke verbinden" zur Auswahl angezeigt. Ohne diese Liste von Server-Hosts müssen Sie den Pfad der Netzressource angeben (siehe "Ferne Dateisysteme bearbeiten")

NFS-Client starten

Wenn Sie bei der Konfiguration des Clients "Enable NFS-Client" selektiert haben, wird der Dienst automatisch beim Starten des Windows aktiviert. Ansonsten können Sie über die Untergruppe "File and Print Services" der Programmgruppe "LAN WorkPlace Pro" den NFS-Client starten.

5.2 Ferne Ressourcen bearbeiten

Bei gestartetem NFS-Client können Sie Dateien und Dateiverzeichnisse des POSIX-Dateisystems, die der NFS-Server für den fernen Client-Zugriff bereitstellt, unter Windows einhängen und bearbeiten. Gehen Sie folgendermaßen vor:

In POSIX (BS2000)

Das Dateisystem, das Sie unter Windows einhängen wollen, stellen Sie für den Client-Zugriff mit folgendem Kommando bereit:

```
share -F nfs ressource
```

ressource ist der vollständige Pfadname der bereitzustellenden Ressource.

Auf dem PC

Die bereitgestellte Ressource des BS2000-Rechners hängen Sie unter Windows in einem virtuellen lokalen Laufwerk ein.

Über die Menüoption "Extras -> Netzlaufwerk verbinden" des Windows-Explorers können Sie ein fernes Dateisystem unter einem lokalen Laufwerks-Kennbuchstaben einhängen.

Der Name des NFS-Servers und des freigegebenen Dateisystems muss als Pfad in der folgenden Form angegeben werden:

```
\\Name_des_NFS-Servers\Dateisystem-Pfad
```

Name_des_NFS-Servers:

Name des BS2000-Rechners. Falls der PC an einem DNS-Service angeschlossen ist, kann hier der Domänenname des BS2000-Hosts angegeben werden. Ansonsten muss der Server-Name entsprechend dem Eintrag in der Hosts-Datei des PCs (auf WinNT: `\winnt\system32\drivers\etc\hosts`) angegeben werden.

Dateisystem-Pfad:

Absoluter Pfad des auf dem Server freigegebenen Dateisystems (so wie er beim *share*-Kommando auf dem Server angegeben wurde).

Anschließend wird ein Dialog zur Angabe von Benutzername und Passwort angezeigt (mit den bei der Installation angegebenen Default-Werten). Hiermit wird eine Berechtigungsprüfung vom *pcnfsd*-Dämon, der auf dem BS2000-Rechner (NFS-Server) läuft, gefordert und die Berechtigung anhand der angegebenen BS2000-Kennung überprüft.



Das Passwort muss in Großbuchstaben angegeben werden.

5.3 Druckumlenkung

Mit NFS (BS2000) und NFS-Client auf PC können Sie lokale Dateien des PCs auf einem BS2000-Drucker ausdrucken. Für diese Druckumlenkung muss auf dem BS2000-Rechner der Dämon *pcnfsd* gestartet sein (wird standardmäßig beim Starten von NFS über das r-Skript gestartet). Zudem muss in der Verwaltungsdatei */etc/print* ein Eintrag für den BS2000-Drucker, der die PC-Dateien ausdrucken soll, vorhanden sein. Gehen Sie folgendermaßen vor:

In POSIX (BS2000)

1. Tragen Sie in der Datei */etc/print* des POSIX-Dateisystems den BS2000-Drucker ein, den Sie vom PC aus ansprechen wollen.

```
printer:::::::::::::
```

printer ist der Aliasname für den BS2000-Drucker, mit dem er vom PC aus angesprochen wird. Die Felder zwischen den Doppelpunkten stehen für Parameter, mit denen das BS2000-PRINT-Kommando versorgt werden kann. Die Angabe dieser Parameter ist wahlfrei.

Beispiel:

```
BS2PRINT::DFSPRINT:'DRUCKUMLENKUNG':(HP):STDWA4:E::(DI,DJ):::
```

Damit haben Sie folgende Werte für das BS2000-PRINT-Kommando festgelegt:

Name des Druckers: BS2PRINT

Standardkonvertierung

Jobname: DFSPRINT

Text für Headerpage: DRUCKUMLENKUNG

HP-Drucker

Papierform: STDWA4

SPACE=E

Laserdrucker (Standard)

Zeichensätze: DI und DJ

Standard-HP-File

2. Stellen Sie das POSIX-Dateiverzeichnis */var/spool* für den fernen Client-Zugriff bereit:

```
share -F nfs /var/spool
```


Auf dem PC

1. Richten Sie einen Netzwerkdrucker ein:

Auf dem WinNT-Desktop erfolgt dies über "Start" -> "Einstellungen" -> "Drucker". Nach Auswahl der Ikone "Neuer Drucker" im daraufhin aktivierten Dialog können Sie über den angebotenen Assistenten einen Druck-Server im Netzwerk definieren. Der danach anzugebende Pfad für den Druck-Server hat folgende Syntax:

```
\\PrinterServer\PrinterName
```

PrinterServer ist der Name des BS2000-Rechners.

PrinterName ist der Name des BS2000-Druckers, wie er in der Datei */etc/print* angegeben wurde.

2. Drucken Sie die Datei aus:

Über das "Drucken.."-Menü der jeweiligen Windows-Applikation können sie den in 1. eingerichteten Netzwerkdrucker auswählen und den Ausdruck anstoßen.

Die zu druckende Datei wird im POSIX-Dateiverzeichnis */var/spool* als Datei mit folgendem Namen abgelegt: *nfs.pcname.zeit*

wobei *pcname* der Name des PCs ist und *zeit* das Datum und die Uhrzeit seit 1.1.1970 in Sekunden angibt.

Der *pcnfsd*-Dämon auf dem BS2000-Rechner überträgt dann die Datei ins DMS (Data Management System) des BS2000 mit dem POSIX-Kommando *bs2cp*. Sie wird unter der Kennung TSOS abgelegt.

Der *pcnfsd*-Dämon auf dem BS2000-Rechner holt sich aus der Datei */etc/print* die Vorlage für das BS2000-PRINT-Kommando. Die Datei wird über SPOOL ausgedruckt.

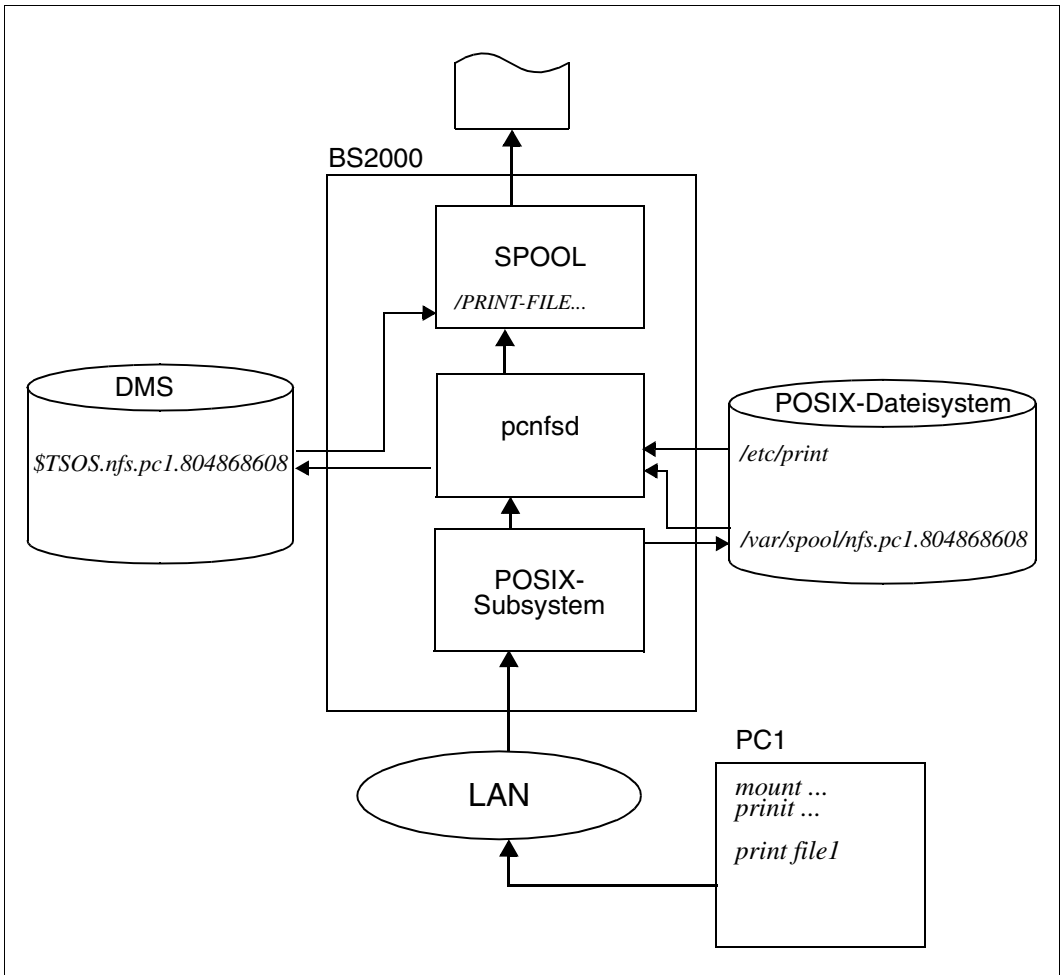


Bild 8: NFS-Druckumlenkung

6 Diagnose und Leistungsverbesserung

Dieses Kapitel gibt einen Einblick in interne Abläufe, beschreibt Probleme, die unter Umständen während des NFS-Betriebs auftreten können und Maßnahmen zur Leistungsverbesserung.

Die in diesem Kapitel enthaltenen technischen Einzelheiten sollen einem erfahrenen Systemverwalter die Diagnose erleichtern. Um sich detailliertere Kenntnisse über NFS anzueignen, können Sie das Buch „Managing NFS and NIS“ zu Rate ziehen.

6.1 Der Einhängvorgang im Überblick

Dieser Abschnitt beschreibt die internen Abläufe beim Start von NFS und beim Einhängen einer fernen Ressource.

Start von NFS

Wenn Sie *nfsstart* eingeben, werden folgende Dämonen gestartet:

```
1 mountd
4 nfsd
4 biod
1 pcnfsd
1 rpcbind
```

Einhängen einer fernen Ressource

Beispielhaft für alle Einhängvorgänge bei fernen Ressourcen wird hier beschrieben, was passiert, wenn Sie das Kommando *mount -F nfs* eingeben.

1. Sie geben folgendes Kommando ein:

```
mount -F nfs -o intr,rw,soft tanz:/usr/src /usr/src/tanz.src
```

2. Es wird überprüft, ob das Kommando *mount* für das Einhängen von fernen Ressourcen im Verzeichnis */etc/fstab* vorhanden ist und ob das eingegebene Kommando den Syntax-Richtlinien entspricht.

3. Das Kommando *mount* prüft in der Datei */etc/mnttab*, ob die Ressource eventuell bereits automatisch eingehängt worden ist.
4. Das Kommando *mount* fordert über den Dämon *rpcbind* die Portnummer an, die der Dämon *mountd* auf dem Rechner *tanz* hat.
5. Das Kommando *mount* übergibt dem Dämon *mountd* auf dem Rechner *tanz* den Pfadnamen der Ressource (*usr/src*).
6. Der Dämon *mountd* auf dem Rechner *tanz* bearbeitet den Einhängauftrag und überprüft, ob die Ressource bereitgestellt wurde und mit welchen Berechtigungen sie bereitgestellt wurde.
7. Das Kommando *mount* erhält eine positive Antwort vom Dämon *mountd* auf dem Rechner *tanz* und fügt in der Datei */etc/mnttab* einen Eintrag für die eingehängte Ressource hinzu.
8. Wenn Sie nun eine Datei der eingehängten Ressource bearbeiten wollen, wird die Anforderung vom Dämon *nfsd* des Rechners *tanz* bearbeitet. Der überprüft anhand der bei der Bereitstellung der Ressource angegebenen Optionen, ob die Anforderung erfüllt werden kann oder abgelehnt werden muss.

6.2 Diagnose

Voraussetzung für die Beseitigung von NFS-Problemen ist ein grundlegendes Verständnis der Umstände, unter denen diese Fehler auftreten können. Bei der Lokalisierung eines NFS-Problems sollte man sich immer bewusst machen, dass der Fehler bei einer der drei Hauptkomponenten aufgetreten sein kann:

- Server
- Client
- Netz

Daher wird empfohlen, diese drei Komponenten bei der Problemsuche möglichst getrennt voneinander zu betrachten, um so die Problemursache leichter lokalisieren zu können.

6.2.1 Server-Probleme

Die Meldungen, die NFS zu Server-Problemen ausgibt, werden auf der BS2000-Konsole ausgegeben. Sie bestehen aus einer POSIX-Meldung, die als &00 die NFS-Meldung enthält:

```
POS1020    MELDUNG DES POSIX-KERNELS: &00
```

Die im folgenden beschriebenen Probleme treten beim Zugriff auf ferne Ressourcen auf. Der Zugriff kann durch Kommandos oder von Programmen aus erfolgen. Beim Zugriff auf Ressourcen des Servers wird zwischen „hart“ eingehängten (hard mounted) und „weich“ eingehängten (soft mounted) Dateiverzeichnissen unterschieden (siehe Kommando *mount* auf Seite 43 unter Option *soft* und *hard*).

Wenn eine Ressource „hart“ eingehängt wurde und der Server aus irgendeinem Grund nicht verfügbar ist, sind alle Programme blockiert, die auf diese Ressource zugreifen wollen. NFS gibt folgende Meldung aus:

```
NFS server <hostname> not responding, still trying
```

Sobald der Server wieder erreichbar ist, erscheint folgende Meldung:

```
NFS server <hostname> ok
```

Es empfiehlt sich in diesem Fall Ressourcen zusätzlich mit der Option *-o intr* einzuhängen, damit hängende Programme abgebrochen werden können.

Wenn eine Ressource „weich“ eingehängt wurde und der Server nicht verfügbar ist, wird folgende Meldung ausgegeben:

```
NFS fstat failed for server <server>: RPC: Timed out
```

In diesem Fall sollten Sie zuerst prüfen, ob der Server aktiv und erreichbar ist. Geben Sie folgendes Kommando ein, wobei *host* der Rechnername des NFS-Servers (<server>) aus der Meldung ist:

```
/usr/bin/rpcinfo host
```

Wenn der Server aktiv ist, wird eine Liste der auf dem Server laufenden Programme mit Versionsnummern, Protokollen und Anschlussnummern ausgegeben. Wenn keine Liste ausgegeben wird, sollten Sie überprüfen, ob der Dämon *rpcbind* auf dem Server läuft.

Wenn der Dämon aktiv, aber für den Client nicht erreichbar ist, sollten Sie die Netzverbindung zwischen Server und Client überprüfen. Auf einem BS2000-Rechner mit POSIX können Sie das auf zwei Arten:

in der POSIX-Shell mit: `ping`

im BS2000-Kommando-Modus mit: `/START-PING`

Wenn der Server wieder verfügbar ist, muss die Ressource wieder neu eingehängt werden.

6.2.2 Client-Probleme

In diesem Abschnitt werden Probleme behandelt, die beim Einhängen von NFS-Ressourcen auftreten können. Jeder Teilschritt des Einhängvorgangs kann zu einem Fehler führen – einige davon sogar zu mehreren. Zu jeder der im folgenden aufgeführten Fehlermeldung sind die möglichen Ursachen angegeben. Die Meldungen, die NFS zu Client-Problemen ausgibt, werden auf `stdout`, also in der POSIX-Shell, ausgegeben.

In den folgenden Beispielen wird davon ausgegangen, dass die Ressource über die Kommandozeile eingehängt wird, die beschriebenen Verfahren zur Fehlersuche gelten jedoch auch für das Einhängen über die Datei */etc/vfstab*.

```
nfs_mount: <server>:<dvz>: server not responding: RPC: program not registered
```

Der Server, der die einzuhängende Ressource bereitstellt, ist nicht aktiv oder der Dämon *rpcbind* läuft nicht. Eine weitere mögliche Fehlerursache ist ein gestörter Netzanschluss des Servers.

```
nfs_mount: <server>:</dvz>: server not responding: RPC:program unavailable
```

Das Kommando *mount* konnte den Dämon *rpcbind* des Servers zwar erreichen, aber der NFS-Dämon *mountd* ist dort nicht registriert. Es ist zu überprüfen, ob die NFS-Dämonen auf dem Server aktiv sind.

```
mount: mount-point does not exist
```

Der lokale Einhängpunkt existiert nicht.

```
nfs_mount: access denied for <server>:/<dvz>
```

Die Ressource wurde nicht bereitgestellt oder der Name Ihres Systems ist in der Liste zugriffsberechtigter Clients nicht enthalten. Überprüfen Sie, ob und wie die einzuhängende Ressource in der Tabelle der bereitgestellten Ressourcen des Servers eingetragen ist. Geben Sie hierzu folgendes Kommando auf dem Client ein, wobei *server* der Rechnername des NFS-Servers (<server>) aus der Meldung ist: `dfshares -F nfs server`
Wenn sie nicht eingetragen ist, so muss die Ressource auf dem Server bereitgestellt werden. Geben Sie hierzu folgendes Kommando auf dem Server ein, wobei *pfadname* der Name der NFS-Ressource (<dvz>) aus der Meldung ist: `share -F nfs pfadname`

```
sh: <datei> cannot create
```

Der Benutzer hat nicht die erforderlichen Zugriffsrechte oder die erforderliche Benutzer-
nummer.

```
mount: ...: Not a directory
```

Bei dem angegebenen Pfadnamen handelt es sich nicht um ein Verzeichnis. Eventuell mit dem Kommando `ls` überprüfen, ob das angegebene Verzeichnis vorhanden ist.

6.3 Maßnahmen zur Leistungsverbesserung

Leistungsprobleme mit NFS können verschiedene Ursachen haben:

- Ungenügende Plattenleistung am Server kann die Durchsatzrate beschränken, mit der man Lese- und Schreibzugriffe ausführen kann.
- Zu hohe CPU-Last des Client oder Servers beschränkt deren Fähigkeit, Netz-Anfragen zu bedienen.
- Ein überlastetes Netz kann die Transferrate beschränken oder die Wiederholungsrate für Anfragen oder Datenübertragungen erhöhen.

Verbesserung des Lese- und Schreibzugriffs

Wird die Datenübertragung für einen NFS-Lese- oder Schreibzugriff nicht erfolgreich abgeschlossen, so wird der gesamte Datenblock noch einmal übertragen. Wenn die Wiederholungsrate zu hoch ist, muss die Größe der NFS-Lese- oder Schreibblöcke reduziert werden. Das ist durch das Setzen der Option *rsize* und *wsize* in der Verwaltungsdatei */etc/vfstab* bzw. beim Einhängen mit dem Kommando *mount* möglich.

Wählen Sie probeweise den Wert 2048 Bytes oder 1024 Bytes beim Eintrag in der Datei */etc/vfstab*:

```
bobserver:/home/bob - /home/bob nfs - - rw,rsize=2048,wsize=2048
```

Da eine Verkleinerung der Lese- und Schreibblöcke die maximal mögliche Leistung beschränkt, kann es auch hilfreich sein, das Zeitlimit für die Anforderungswiederholungen herabzusetzen. Setzen Sie die Option *timeo* beim Kommando *mount* oder in der Datei */etc/vfstab* auf 8 oder 6 Zehntelsekunden (der Standardwert beträgt 11).

Verbesserung der Netzleistung

Wenn sich ein eingehängtes NFS-Dateisystem über ein weites Netz mit vielen Gateways oder großen Entfernungen zwischen Clients und Server erstreckt, kann es notwendig sein, die Option *timeo* beim Kommando *mount* zu erhöhen. Satellitenverbindungen oder andere Netz-Hardware können lange Verzögerungen auf dem Übertragungsweg bewirken. Das Zeitlimit muss in diesem Fall erhöht werden. Setzen Sie *timeo* auf 50 oder 100 Zehntelsekunden.

Benutzen Sie *nfsstat -rc*, um die den Client betreffende Anzahl der NFS-/RPC-Übertragungswiederholungen und Fehler zu prüfen.

Fachwörter

bereitstellen

Freigabe von lokalen Ressourcen zum Einhängen auf fernen Rechnern; wird auch exportieren genannt.

Client

Im Zusammenhang mit NFS ist ein Client der Rechner, der auf Ressourcen zugreift, die von einem anderen Rechner (Server) bereitgestellt wurden.

Behälterdatei

physikalisches Speichermedium für ein Dateisystem des POSIX-Dateibaums. Eine Behälterdatei ist eine PAM-Datei, die sich auf einem PVS (Public Volume Set) befindet.

Dämon

Dämonen sind Systemprozesse, die permanent und meist im Hintergrund laufen und die allgemeine Aufgaben durchführen.

Dateibaum

Gesamte Hierarchie der Dateien auf einem UNIX-Rechner bzw. in POSIX; sie wird Dateibaum genannt, da sie gemäß einer Baumstruktur geordnet ist. Die Wurzel des Dateibaums ist das Root-Dateiverzeichnis (/). Alle anderen Dateiverzeichnisse sind Zweige, die vom Root-Dateiverzeichnis ausgehen. Die Dateien sind die Blätter des Baumes. Jede Datei ist über genau einen Pfad des Dateibaums erreichbar.

Dateisperre

Prozesse, die auf gemeinsam benutzten Dateien arbeiten, synchronisieren den Zugriff auf solche Dateien über Sperren. Das Arbeiten mit Sperren in NFS wird in einem gesonderten Protokoll „NLM“ (Network Lock Manager) realisiert.

Dateisystem

Ein Dateisystem ist eine hierarchische Sammlung von Dateiverzeichnissen und Dateien, die sich physikalisch auf demselben Speichermedium befinden, z.B. auf einer Partition oder in einer Behälterdatei.

Der Begriff wird für Organisationsformen von Dateien benutzt: wie z.B. UNIX-Dateisystem, POSIX-Dateisystem, hierarchisches Dateisystem, andere BS2000-Dateisysteme (z.B. DMS, Data Management System).

Dateisystem-Typ

Art eines Dateisystems im Dateibaum von POSIX oder eines UNIX-Rechners. Die bekanntesten Typen sind:

Typ *ufs*: lokale Dateisysteme mit Benutzerdaten.

Typ *nfs*: Dateisysteme, die sich physikalisch auf fernen Rechnern befinden.

Weitere Typen sind z.B. *dfs*, *proc*, *bfs*, *rfs* oder *s5*.

Dateiverzeichnis

Ein Dateiverzeichnis wird verwendet, um Dateien und untergeordnete Dateiverzeichnisse eines hierarchischen Dateisystems zu gruppieren und zu organisieren.

einhängen

logische, nicht physikalische Übernahme einer fernen Ressource in den lokalen Dateibaum; wird auch montieren genannt.

ferner Rechner

Ein Rechner in einem Rechnernetz, an dem der Benutzer nicht direkt arbeitet. Der Rechner, an dem der Benutzer direkt arbeitet, wird als lokaler Rechner bezeichnet. Der Benutzer kann mit fernen Rechnern im Netz kommunizieren.

heterogene Rechnerumgebung

Ein Rechnernetz, in dem Rechner von unterschiedlichen Herstellern und mit verschiedenen Betriebssystemen miteinander kommunizieren. Synonym: Offenes Rechnernetz.

IP (Internet Protocol)

Das Internet Protocol ist ein Protokoll, das die Wegewahl in einem Rechnernetz durchführt. Es leistet einen Teil der Funktionen, die von der Schicht 3 des ISO-Referenzmodells gefordert werden. Auf IP setzen die Protokolle TCP und UDP auf.

LAN (Local Area Network)

Ein LAN ist ein Rechnernetz, dessen räumliche Ausdehnung auf ein bestimmtes Gebiet beschränkt ist. In der Bundesrepublik Deutschland ist die Größe auf das Grundstück des Anwenders beschränkt. Ein LAN kann als privates Subnetz mit anderen Rechnernetzen verbunden sein und so Teil eines größeren Netzes, etwa eines WAN sein. Synonym: lokales Rechnernetz, lokales Netz.

lokaler Rechner

Der Rechner, an dem der Benutzer direkt arbeitet, wird als der lokale Rechner bezeichnet. Synonym: Host.

OSI-Referenzmodell

Grundlage der ISO-Normierung der Datenkommunikation ist das Modell für die Kommunikation offener Systeme, das OSI-Referenzmodell (Open System Interconnection). Das OSI-Modell strukturiert den Aufbau von Kommunikationssystemen und schafft die Basis für die Normung der Protokolle und Dienste. Es schreibt fest, welche Funktionen die an der Kommunikation beteiligten Komponenten erbringen müssen.

Das OSI-Referenzmodell besteht aus sieben hierarchisch aufeinandergelegten Schichten. Jeder Schicht sind im Rahmen der Kommunikationsaufgabe spezifische Funktionen zugeordnet.

Portnummer

Eine Portnummer ermöglicht es, eine bestimmte Anwendung innerhalb eines Rechners zu adressieren. Sie entspricht der Adresse einer Anwendung in einem Rechner. Die Kombination Internet-Adresse/Portnummer identifiziert den Empfänger bzw. Sender eines Datenpaketes innerhalb des Netzes eindeutig.

POSIX-Dateisystem

Dateisystem auf einem BS2000-Rechner mit POSIX. Das POSIX-Dateisystem entspricht einem UNIX-Dateisystem.

PVS (Public Volume Set)

Gruppe gemeinschaftlicher Datenträger (max. 16) mit oder ohne Systemdateien. Wird auch Pubset genannt.

Ressource

Dateien oder Dateiverzeichnisse, die mit NFS benutzt werden.

Root-Dateiverzeichnis

Das Dateisystem, an dem der Dateibaum beginnt. Es wird auch Wurzel-Dateiverzeichnis genannt und durch den Schrägstrich (/) dargestellt.

RPC (Remote Procedure Call)

Der entfernte Prozeduraufruf ist das Verfahren, das für die Kommunikation zwischen Client und Server bei verteilter Verarbeitung benutzt wird. Es wird von NFS verwendet. Die Identifikation eines Server-Programms erfolgt über eine Programm-, eine Prozedur- und eine Versionsnummer. RPC setzt auf die Protokolle TCP und UDP auf. Für NFS wird nur UDP benutzt.

Server-Rechner

Im Zusammenhang mit NFS ist ein Server der Rechner, der Ressourcen bereitstellt, die von anderen Rechnern (Clients) eingehängt und bearbeitet werden können.

SYSLST

BS2000-Systemdatei für druckaufbereitete Listen.

TCP (Transmission Control Protocol)

TCP ist ein verbindungsorientiertes Protokoll, das den Datentransport zwischen zwei Rechnern abwickelt. TCP gewährleistet (im Gegensatz zum UDP) eine gesicherte Datenübertragung (End-to-End-Verbindung) und ist Schicht 4 des ISO-Referenzmodells zuzurechnen.

transparenter Dateizugriff

Der Benutzer kann auf Dateien eines fernen Rechners wie auf die Dateien des lokalen Rechners zugreifen.

UDP (User Datagram Protocol)

UDP ist ein verbindungsloses Protokoll, welches den Datentransport zwischen zwei Rechnern abwickelt. UDP ist etwa auf Schicht 4 des ISO-Referenzmodells anzusiedeln. Das UDP ist ein Datagramm-Protokoll, das Broadcasting unterstützt. Im Gegensatz zu TCP (gesichertes End-zu-End-Protokoll) gibt UDP nur die Gewähr, dass die Nachricht erfolgreich gesendet wurde.

UFS (UNIX File System)

Lokales Dateisystem des UNIX.

WAN (Wide Area Network)

Unter einem WAN versteht man ein Rechnernetz, das nicht auf ein räumlich begrenztes Gebiet beschränkt ist.

Literatur

Literatur zu NFS

Managing NFS and NIS

Hal Stern

O'Reilly&Associates, Inc

ISBN 0-937175-75-7

Reliant UNIX V5.44

Netzwerke - Leitfaden für Benutzer und Verwalter

Benutzerhandbuch

Reliant UNIX V5.44

Netzverwaltung

Systemverwalterhandbuch

Sonstige Literatur

BCAM Band 1 (BS2000/OSD)

Benutzerhandbuch

BCAM Band 2 (BS2000/OSD)

Referenzhandbuch

BS2000/OSD-BC V4.0

Einführung in die Systembetreuung

Benutzerhandbuch

BS2000/OSD-BC V4.0

Kommandos Band 1 bis Band 6

Benutzerhandbuch

EDT (BS2000/OSD)

Anweisungen

Benutzerhandbuch

PDN

Benutzerhandbuch

POSIX-BC (BS2000/OSD)

Grundlagen für Anwender und Systemverwalter

Benutzerhandbuch

POSIX-BC (BS2000/OSD)

Kommandos

Benutzerhandbuch

SPOOL (BS2000/OSD)

Benutzerhandbuch

SPOOL (BS2000/OSD)

SPOOL & Print-Kommandos

Benutzerhandbuch

Wenden Sie sich zum Bestellen von Handbüchern bitte an Ihre zuständige Geschäftsstelle.

Stichwörter

/etc/dfs/dfstab 90
/etc/dfs/fstypes 91
/etc/dfs/sharetab 92
/etc/mnttab 82
/etc/print 83
 bearbeiten 96
/etc/rmtab 85
/etc/rpc 86
/etc/vfstab 88

64-Bit-Dateisystem 12

A

ASCII 32
asynchrones Schreiben 9
Ausgeben von RPC-Informationen 76
aushängen 35

B

BCAM 13
beenden
 NFS 30
 POSIX-Shell 29
Behälterdatei 14, 31
 Zugriffsschutz 20
Benutzerverwaltung 19
bereitstellen 7, 33
 mehrerer Ressourcen 55
Bereitstellung zurücknehmen 33, 62
 mehrere Ressourcen zurücknehmen 63
biod 66
BS2000-Datei 33

C

Cache 8
Cache-Puffer 45
Client 7
Client-Dämon 66
Client-Probleme 102
Code-Konvertierung 32

D

Dämon für
 Anforderungen von PC-Clients 72
 Anforderungen zum Einhängen ferner
 Ressourcen 70
 RPC 73
Dämonen
 biod 66
 lockd-clnt 67
 lockd-srv 68
 mountd 57, 70
 nfsd 71
 pcnfsd 72
 rpcbind 73
 statd 74
Dämonen starten 64
Dämonen überwachen 65
Dämonen, Übersicht 64
Darstellungsmittel 3
Datei 6
Dateisperre 9, 68
Dateisystem 6, 35
Dateiverzeichnis 6
Dateizugriffsschutz 20
DCAM 13
dfmounts 39

Diagnose

Client 102
Server 101

E

EBCDIC 14, 32
eingehängte Dateisysteme an fernen Rechnern anzeigen 57
einhängen 7, 34
auf dem PC 95
hart oder weich 45, 101
im Vorder- oder Hintergrund 45
Einhängepunkt 5, 7
Einhängevorgang 99
exportieren 33

F

ferne Ressourcen aushängen 60
ferne Ressourcen einhängen 43

H

hart einhängen 45, 101

I

IMON 25
Informationen über eingehängte Ressourcen ausgeben 39
Informationsdatei 26
Installation
NFS 25
POSIX 26
Installation von DFS (MS-DOS) 95
Installations Monitor 25
Installationsprogramm 26

K

Kommandos
dfmounts 39
dfshares 41
mount 43
mountall 47
nfsstat 48
share 53
shareall 55

showmount 57
umount 60
umountall 61
unshare 62
unshareall 63

Kommandos eingeben 37
Kommandos, Übersicht 37

L

Leistungsverbesserung
Lese-/Schreibzugriff 104
Netzleistung 104
Lesevorgang 9
Liefereinheit
Dateien 25
lockd-clnt 10, 67
lockd-srv 10, 68
lokale Ressourcen für Client-Zugriffe bereitstellen 53

M

mehrere ferne Ressourcen aushängen 61
mehrere ferne Ressourcen einhängen 47
montieren 34
mount 43
mountall 47
mountd 70

N

Network Lock Manager 9, 10
Network Status Monitor 10, 74
Netzanbindung 13
NFS
Dateien > 2 GB 12
starten und beenden 30
Verwaltung 29
NFS-Cache 8
NFS-Client-Dämon 66
nfsd 71
NFS-Kommandos 15

- NFS-Meldung
 - auf Konsole 101
 - in POSIX-Shell 102
- NFS-Server-Dämon 71
- nfsstat 48
- NFS-Versionen 8
- NLM-Client-Dämon 67
- NLM-Protokoll 10
- NML-Server-Dämon 68
- NSM-Protokoll 10

- P**
- PAM-Datei 14, 31
- Portnummer, privilegierte 23
- Portüberwachung 23
- POSIX 13
 - Informationsdatei 23
 - Kommandos 14
 - Programmpaket 13
- POSIX-Benutzerverwaltung 19
- POSIX-Datei
 - Code-Konvertierung 32
 - Namenskonventionen 32
- POSIX-Dateisystem 14, 35
 - anlegen 31
 - speichern 31
- POSIX-Programmpaket 26
- POSIX-Shell 13
 - starten und beenden 29
- POSIX-Subsystem 14, 26
- Protokoll
 - verbindungslos 8

- R**
- Ressourcen 6
 - automatisch einhängen 36
 - bereitstellen 33, 92
 - einhängen 34
 - informieren über 36
- Root-Berechtigung 20
- RPC 16, 75
 - rpcbind 73
 - RPC-Dienst
 - zustandsbehaftet 74
 - rpcinfo 76
 - RPC-Programmnummer-Datei 86

- S**
- Schablonen für BS2000-Druckkommandos 83
- Schreibmodus
 - asynchron 9
 - synchron 8
- Schreibvorgang 8
- Schutzbits 21
- Schutzfrist für Dateisperren 68
- Server 7
 - Server-Dämon 71
 - Server-Probleme 101
- share 53
- shareall 55
- SOLIS 25
- starten
 - Dämonen 64
 - NFS 30
 - POSIX-Shell 29
- statd 10, 11, 74
- statistische Informationen ausgeben 48
- Status-Dämon 74
- Status-Monitor 11

- T**
- Tabelle aller definierten Dateisysteme 88
- Tabelle der bereitgestellten Ressourcen 92
- Tabelle der bereitzustellenden Ressourcen 90
- Tabelle der installierten Utilities für verteilte Dateisysteme 91
- TCP/IP 13
- TPR-Teile von NFS 15
- transparenter Zugriff 7

- U**
- UDP 8
- umount 60
- umountall 61
- unshare 62
- unshareall 63
- User Datagram Protocol 8

V

verfügbare ferne Ressourcen auflisten 41

Verwaltungsdateien

 /etc/dfs/dfstab 90

 /etc/dfs/sharetab 92

 /etc/mnttab 82

 /etc/print 83

 /etc/rmtab 85

 /etc/rpc 86

 /etc/vfstab 88

Verwaltungsdateien bearbeiten 80

Verwaltungsdateien, Übersicht 80

vfstab 36

W

weich einhängen 45, 101

Wiederanlaufverhalten 10

Z

Zugriffsschutz

 Behälterdateien 20

 ferne Ressourcen 22

 POSIX-Dateien 21

Inhalt

1	Einleitung	1
1.1	Kurzbeschreibung des Produkts	1
1.2	Zielgruppen des Handbuchs	1
1.3	Wegweiser durch das Handbuch	2
1.4	Readme-Datei	2
1.5	Darstellungsmittel	3
1.6	Änderungen gegenüber Vorgängerhandbuch	4
2	Überblick und Einbettung im BS2000	5
2.1	Überblick über NFS	5
2.1.1	Arbeiten mit verteilten Dateisystemen	5
2.1.2	NFS-Versionen und NFS-Protokollversionen	8
2.1.3	Der Schreib- und Lesevorgang bei NFS	8
2.1.4	Der Network Lock Manager	9
2.1.5	Der Status-Monitor	11
2.1.6	Dateisystemgrößen ab 2 Gbyte	12
2.2	NFS im BS2000/OSD	13
2.2.1	Netzanbindung	13
2.2.2	POSIX	13
2.2.3	Bestandteile von NFS	15
2.2.4	Zusammenwirken von NFS und POSIX	17
2.3	Sicherheit	19
2.3.1	Benutzerverwaltung	19
2.3.2	Dateizugriffsschutz	20
2.3.3	Portüberwachung	23
3	Installation und Einsatz	25
3.1	Installation von NFS	25
3.2	POSIX-Shell starten und beenden	29
3.3	Einsatz von NFS	29
3.3.1	NFS starten und beenden	30
3.3.2	Besonderheiten des POSIX-Dateisystems	31
3.3.2.1	Code-Konvertierung	32
3.3.2.2	BS2000-Dateien	33

3.3.3	Ressourcen bereitstellen und Bereitstellung zurücknehmen	33
3.3.4	Ressourcen ein- und aushängen	34
3.3.5	Informieren über Ressourcen	36
4	Kommandos, Dämonen, Verwaltungsdateien	37
4.1	NFS-Kommandos	37
	dfmounts Über eingehängte Ressourcen informieren	39
	dfshares Über bereitgestellte Ressourcen informieren	41
	mount Ferne Ressourcen einhängen	43
	mountall Mehrere ferne Ressourcen einhängen	47
	nfsstat Statistische Informationen ausgeben	48
	share Lokale Ressourcen für Client-Zugriffe bereitstellen	53
	shareall Mehrere Ressourcen für den Client-Zugriff bereitstellen	55
	showmount Über NFS-Clients und Ressourcen informieren	57
	umount Ferne Ressourcen aushängen	60
	umountall Mehrere ferne Ressourcen aushängen	61
	unshare Bereitstellung von Ressourcen zurücknehmen	62
	unshareall Bereitstellung mehrerer Ressourcen zurücknehmen	63
4.2	Dämonen	64
	biod NFS-Client-Dämon für blockorientierte Ein-/Ausgabe	66
	lockd-clnt Dämon für NLM-Clients	67
	lockd-srv RPC-Dienst für NLM (Network Lock Manager)	68
	mountd Dämon zum Einhängen ferner Ressourcen	70
	nfsd NFS-Server-Dämon für Ein-/Ausgabe	71
	pcnfsd Dämon zur Unterstützung von DOS-PCs	72
	rpcbind Dämon für RPC	73
	statd Status-Monitor für zustandsbehaftete RPC-Dienste	74
4.3	rpcinfo-Programm	75
	rpcinfo Ausgeben von RPC-Informationen	76
4.4	Verwaltungsdateien	80
	/etc/mnttab Tabelle der eingehängten Dateisysteme	82
	/etc/print Schablonen für BS2000-Druckkommandos	83
	/etc/rmtab Tabelle der eingehängten fernen Ressourcen	85
	/etc/rpc RPC-Programmnummern-Datei	86
	/etc/vfstab Tabelle der definierten Dateisysteme	88
	/etc/dfs/dfstab Tabelle der bereitzustellenden Ressourcen	90
	/etc/dfs/fstypes Tabelle der installierten Utilities für verteilte Dateisysteme	91
	/etc/dfs/sharetab Tabelle der bereitgestellten Ressourcen	92
5	Anbindung eines Windows-PCs	93
5.1	Vorbereitungen	93
5.2	Ferne Ressourcen bearbeiten	95
5.3	Druckumlenkung	96

6	Diagnose und Leistungsverbesserung	99
6.1	Der Einhängvorgang im Überblick	99
6.2	Diagnose	101
6.2.1	Server-Probleme	101
6.2.2	Client-Probleme	102
6.3	Maßnahmen zur Leistungsverbesserung	104
	Fachwörter	105
	Literatur	109
	Stichwörter	111

NFS V3.0 / NFS V1.2C (BS2000/OSD)

Network File System

Zielgruppe

NFS-Benutzer und NFS-Verwalter

Inhalt

Das Handbuch beschreibt, wie Sie NFS auf einem BS2000-Rechner mit POSIX betreiben und nutzen. Es enthält einen Überblick über verteilten Dateizugriff mit NFS und beschreibt die Kommandos, Dämonen, Verwaltungsdateien und die Fehlerbehandlung.

Ausgabe: Februar 2001

Datei: nfs.pdf

Copyright © Fujitsu Siemens Computers GmbH, 2001.

Alle Rechte vorbehalten.

Liefermöglichkeiten und technische Änderungen vorbehalten.

Alle verwendeten Hard- und Softwarenamen sind Handelsnamen und/oder Warenzeichen der jeweiligen Hersteller

Fujitsu Siemens Computers GmbH
Handbuchredaktion
81730 München

Kritik Anregungen Korrekturen

Fax: 0 700 / 372 00000

e-mail: manuals@fujitsu-siemens.com
<http://manuals.fujitsu-siemens.com>

Absender

Kommentar zu NFS V3.0 / NFS V1.2C
Network File System



Information on this document

On April 1, 2009, Fujitsu became the sole owner of Fujitsu Siemens Computers. This new subsidiary of Fujitsu has been renamed Fujitsu Technology Solutions.

This document from the document archive refers to a product version which was released a considerable time ago or which is no longer marketed.

Please note that all company references and copyrights in this document have been legally transferred to Fujitsu Technology Solutions.

Contact and support addresses will now be offered by Fujitsu Technology Solutions and have the format ...@ts.fujitsu.com.

The Internet pages of Fujitsu Technology Solutions are available at [http://ts.fujitsu.com/...](http://ts.fujitsu.com/) and the user documentation at <http://manuals.ts.fujitsu.com>.

Copyright Fujitsu Technology Solutions, 2009

Hinweise zum vorliegenden Dokument

Zum 1. April 2009 ist Fujitsu Siemens Computers in den alleinigen Besitz von Fujitsu übergegangen. Diese neue Tochtergesellschaft von Fujitsu trägt seitdem den Namen Fujitsu Technology Solutions.

Das vorliegende Dokument aus dem Dokumentenarchiv bezieht sich auf eine bereits vor längerer Zeit freigegebene oder nicht mehr im Vertrieb befindliche Produktversion.

Bitte beachten Sie, dass alle Firmenbezüge und Copyrights im vorliegenden Dokument rechtlich auf Fujitsu Technology Solutions übergegangen sind.

Kontakt- und Supportadressen werden nun von Fujitsu Technology Solutions angeboten und haben die Form ...@ts.fujitsu.com.

Die Internetseiten von Fujitsu Technology Solutions finden Sie unter [http://de.ts.fujitsu.com/...](http://de.ts.fujitsu.com/), und unter <http://manuals.ts.fujitsu.com> finden Sie die Benutzerdokumentation.

Copyright Fujitsu Technology Solutions, 2009