

# 1 Einleitung

Unter **POSIX** (**P**ortable **O**pen **S**ystem **I**nterface for **U**NIX) versteht man eine Reihe von Standards auf UNIX-Basis. Diese Standards gewährleisten die Kompatibilität und Interoperabilität von Anwendungen in einem heterogenen Netz. Ein heterogenes Netz besteht aus Rechnern von verschiedenen Herstellern sowie aus System- und Anwendersoftware von verschiedenen Softwareanbietern.

Der POSIX-Standard wurde vom Institute of Electrical and Electronics Engineers (IEEE) 1989 als nationaler amerikanischer Standard definiert. Anschließend wurde er vom X/OPEN-Konsortium erweitert und 1990 als internationaler Standard verabschiedet (X/OPEN Portability Guide IV).

Der X/OPEN Portability Guide IV, kurz auch XPG4-Standard genannt, umfaßt 7 Bände, die u.a. Schnittstellendefinitionen zu Basisbetriebssystemen, Programmiersprachen, Datenverwaltung und Vernetzung enthalten. Das Betriebssystem BS2000/OSD unterstützt ab V2.0 die XPG4-Standards, die in den ersten beiden Bänden enthalten sind:

- Band 1: System Interfaces and Headers (ca. 350 Programmschnittstellen)
- Band 2: Commands and Utilities (ca. 200 Benutzerschnittstellen)

Zur Unterstützung dieser Schnittstellen hat Siemens Nixdorf die POSIX-Funktionalität im BS2000/OSD integriert. POSIX bezeichnet also sowohl den Standard vom IEEE als auch die BS2000/OSD-Funktionalität „POSIX“. Mit POSIX wurden die Voraussetzungen für eine erfolgreiche Zertifizierung nach dem XPG4-Standard geschaffen, die in zwei Stufen erfolgte: Ende 1995 erhielt BS2000/OSD von „The Open Group“ (vormals X/OPEN) das „XPG4 Base Branding“ (XPG4) und Mitte 1997 das Branding nach „XPG4 UNIX profile“ (auch XPG4.2 oder UNIX95 genannt).

Der Kern der POSIX-Funktionalität ist als privilegiertes BS2000-Subsystem realisiert. Dem Benutzer stehen die Bibliotheksfunktionen des XPG4-Standards über eine C-Bibliothek und eine definierte Menge von Kommandos über eine Shell (POSIX-Shell) zur Verfügung. Die C-Bibliothek ist Bestandteil des Produkts CRTE (Common Runtime Environment).

Mit POSIX lassen sich Anwendungsprogramme leicht portieren - unabhängig vom ausführenden Betriebssystem. Deshalb können XPG4-konforme Programme nach einer Neuübersetzung auch im BS2000/OSD ablaufen.

Die POSIX-Programmschnittstellen werden parallel zu den BS2000-Programmschnittstellen angeboten. Die gemischte Nutzung von BS2000- und POSIX-Programmschnittstellen in einem Programm ist mit Einschränkungen möglich.

## 1.1 Konzept der POSIX-Dokumentation

Für das Kennenlernen von POSIX und das Arbeiten mit dem Subsystem POSIX und der POSIX-Shell im BS2000/OSD steht Ihnen folgende Dokumentation zur Verfügung:

- Einen Überblick über die Strategie und die Ziele von POSIX im BS2000/OSD bietet Ihnen die Broschüre „POSIX im BS2000/OSD“.
- Eine Einführung in das Arbeiten mit dem Subsystem POSIX erhalten Sie im vorliegenden Handbuch „POSIX - Grundlagen für Anwender und Systemverwalter“. Darüber hinaus werden die Verwaltungsaufgaben beschrieben, die im Zusammenhang mit dem Subsystem POSIX anfallen. Außerdem erfahren Sie, mit welchen BS2000-Softwareprodukten Sie das Subsystem POSIX nutzen können.
- Eine vollständige Beschreibung der POSIX-Kommandos, mit denen Sie in der POSIX-Shell arbeiten können, enthält das Handbuch „POSIX-Kommandos“ [2].

### POSIX-Dokumentation im BS2000/OSD-Umfeld

Im BS2000/OSD werden Softwareprodukte funktionell erweitert, so daß Sie auch mit diesen Produkten die POSIX-Funktionalität nutzen können.

Eine Reihe von Dienstprogrammen ermöglichen den Zugriff auf das POSIX-Dateisystem. So können Sie z.B. mit EDT ab V16.5 Dateien des POSIX-Dateisystems bearbeiten.

Durch die Erweiterung des CRTE gemäß dem XPG4-Standard können Sie mit den C-Bibliotheksfunktionen unabhängig vom ausführenden Betriebssystem portable C-Programme schreiben.

Als Grundlage für den Zugriff auf die POSIX-Funktionalität aus anderen Softwareprodukten wird das vorliegende Handbuch „POSIX - Grundlagen für Anwender und Systemverwalter“ vorausgesetzt.

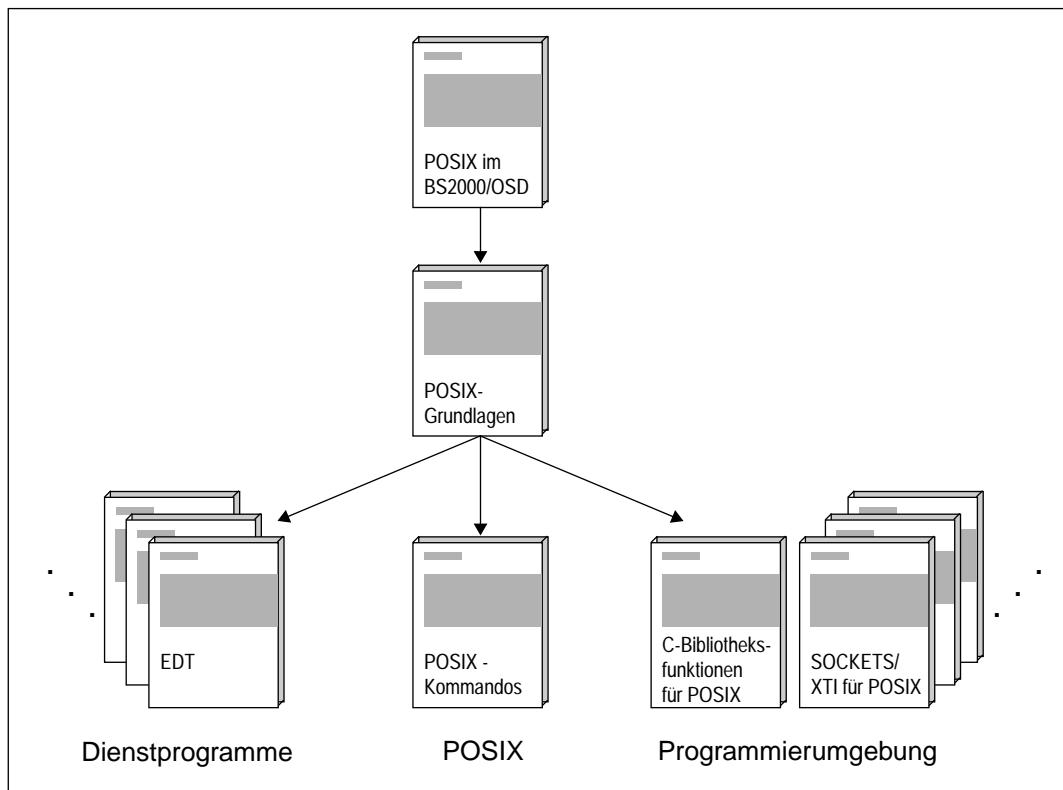


Bild 1: POSIX-Dokumentation im BS2000-Umfeld

## **1.2 Zielgruppen des Handbuchs**

Dieses Handbuch wendet sich an:

- DV-Organisatoren, die einen Überblick über POSIX gewinnen möchten.
- Nichtprivilegierte BS2000-Benutzer, die mit POSIX arbeiten wollen. Grundlegende Kenntnisse des Betriebssystems SINIX/Reliant UNIX sind von Vorteil.
- Benutzer von Workstations, die bisher überwiegend mit SINIX/Reliant UNIX gearbeitet haben und nun POSIX nutzen möchten. BS2000-Grundkenntnisse sind erforderlich.
- BS2000-Systemverwalter und POSIX-Verwalter. Gute Kenntnisse der Betriebssysteme BS2000/OSD und SINIX/Reliant UNIX sind erforderlich.

Grundbegriffe des Betriebssystems SINIX/Reliant UNIX sind in diesem Handbuch für BS2000-Benutzer mit aufgenommen.

## **1.3 Konzept des Handbuchs**

Dieses Handbuch enthält Kapitel, die für alle Benutzer wichtig sind, und Kapitel, die nur für BS2000-Systemverwalter, POSIX-Verwalter oder BS2000-Gruppenverwalter von Bedeutung sind.

Die Kapitel 1 bis 4 wenden sich an alle Benutzer:

- Einleitung
- Einführung in POSIX
- Arbeiten mit POSIX
- BS2000-Softwareprodukte im Umfeld von POSIX

Die Kapitel 5 bis 7 wenden sich an BS2000-Systemverwalter, POSIX-Verwalter und BS2000-Gruppenverwalter:

- POSIX installieren
- POSIX steuern, Dateisysteme verwalten
- POSIX-Benutzer verwalten

Im Referenzteil dieses Handbuchs (Kapitel 8 bis 10) finden Sie:

- die BS2000-Kommandos für POSIX
- die POSIX-Meldungen
- die Privilegien bei POSIX
- den Kommandoumfang der Basis-Shell
- die Dateiverzeichnisse, die bei einer POSIX-Erstinstallation angelegt werden
- die Gerätedateien, die bei einer POSIX-Erstinstallation angelegt werden
- die Verwaltungsdateien, die bei einer POSIX-Erstinstallation angelegt werden
- die SDF-Syntaxbeschreibung
- die Kommando-Returncodes

Im Anschluß an den Referenzteil finden Sie verschiedene Verzeichnisse, die Ihnen das Arbeiten mit diesem Handbuch erleichtern.

### **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)
```

## 1.4 Änderungen gegenüber der vorigen Ausgabe

Diese Handbuchausgabe enthält gegenüber dem Vorgängerhandbuch „POSIX V1.0A (BS2000/OSD), Grundlagen für Anwender und Systemverwalter“, Ausgabe Juli 1995, folgende Änderungen:

- Die Einleitung enthält einen neuen Abschnitt zu eventuell verfügbaren Readme-Dateien und Hinweise auf die zweistufige Zertifizierung nach dem XPG4-Standard.
- Das Betriebssystem SINIX wurde mit der Version 5.43 in Reliant UNIX umbenannt. Diese Namensgebung ist die Konsequenz aus der Zusammenführung der UNIX-Versionen von Siemens Nixdorf und Pyramid Technology. Der Name Reliant UNIX steht für die hohen Anforderungen an Zuverlässigkeit und Verfügbarkeit, die dieses standardisierte Betriebssystem erfüllt – im kommerziellen wie im technischen Einsatz – wobei alle bewährten SINIX-Eigenschaften in Reliant UNIX erhalten bleiben.
- Im Abschnitt „POSIX und das World Wide Web (WWW)“ auf Seite 18 ist die WWW-Serversoftware *WebServe* vorgestellt, die ab BS2000/OSD-BC V2.0 verfügbar ist.
- Der Abschnitt „r-Kommandos“ auf Seite 31 ist neu.
- Der neue POSIX-Lader ist auf Seite 51 beschrieben.
- Das Beispiel auf Seite 56 ist aktualisiert.
- Die Liste der Produkte im Umfeld von POSIX wurde erweitert. Dazugekommen sind:
  - AID
  - SORT
  - TCP-IP-AP
  - TCP-IP-SV
  - DCE
  - TransView SNMP-Basic-Agent BS2000
- POSIX-Installation:
  - Die vollständige Liste der Lieferbestandteile ist nicht mehr im Handbuch enthalten. Sie finden die Liste in der Freigabemitteilung.
  - Das SSC-Objekt SYSSSC.POSIX-BC.<version> (Subsystem POSIX) muß nicht mehr in den Subsystemkatalog eingetragen werden, sondern es wird bei der Installation automatisch übernommen.
  - Die Meldungsdatei muß nicht mehr explizit aktiviert werden – sie wird bei der Installation in einen der Meldungscontainer eingemischt.
  - Die Liste mehrfach benutzbaren Dateien wurde aktualisiert.
  - Der Abschnitt „Liefer- und Installationsverfahren für POSIX-Programmpakete“ ab Seite 81 enthält folgende neue Beschreibungen:
    - Installation aus dem Software Configuration Inventory (SCI)
    - Installation von einer privaten Ablagekennung
    - Protokollierung der Installation

- Einen Hinweis zur Installation von Web*Serve* finden Sie auf Seite 85.
- Die neue Installationsvariante „Upgrade-Installation“ ist auf Seite 86 beschrieben.
- Die Maske „BS2000 POSIX package installation“ (Seite 92) ist geändert.
- Die POSIX-Informationsdatei enthält die neuen Parameter DBLPOOL, DBLSTATE und MAXTIMERC. Der Wert des Parameters MAXUP wurde geändert.
- Bei den BS2000-Kommandos für POSIX (ab Seite 127) wurde der SDF-Datentyp <posix-pathname 1..1024> generell durch <posix-pathname 1..1023> ersetzt.

## 1.5 Verwendete Metasprache

In den Kommando-/Anweisungsformaten werden bestimmte Zeichen und Darstellungsformen verwendet. Ihre Bedeutung ist im Abschnitt „SDF-Syntaxbeschreibung“ auf Seite 186 erläutert.

Folgende Konventionen gelten für die Darstellungsmittel im Text:

- Im Fließtext ist nicht zwischen Konstanten und Variablen unterschieden. Alle Elemente der Syntax, Teile aus Datenstrukturen sowie Dateinamen, Pfadnamen und Kommandos sind dort in *kursiver* Schrift dargestellt.
- In Anwendungsbeispielen sind Eingaben in das System in Schreibmaschinenschrift dargestellt. Alle Eingabezeilen werden bei Zeichenterminals mit der Taste **↵** abgeschlossen, bei Blockterminals mit **[EM] [DUE]**. Deshalb sind die Tasten am Ende der Eingabezeilen weggelassen.

Manche Eingaben sind terminalabhängig, d.h. sie unterscheiden sich bei Block- und Zeichenterminals (siehe dazu auch den Abschnitt „Unterstützung von Terminals“ auf Seite 21).

Ausgaben des Betriebssystems sind in Schreibmaschinenschrift dargestellt.

- Literaturhinweise sind im Text durch Kurztitel angegeben, die in Anführungszeichen stehen. Die vollständigen Titel finden Sie zusammen mit einer Kurzbeschreibung im Literaturverzeichnis.
- Verweise innerhalb dieses Handbuchs geben die betreffende Seite im Handbuch an und je nach Bedarf auch den Abschnitt bzw. das Kapitel. Verweise auf Themen, die in einem anderen Handbuch beschrieben sind, enthalten nur den Kurztitel dieses Handbuchs. Über das Stichwortverzeichnis können Sie in dem genannten Handbuch dann die entsprechende Stelle im Text finden.



Dieses Symbol steht vor Warnungen, die Sie im Interesse der System- und Betriebssicherheit unbedingt beachten müssen.



Dieses Symbol kennzeichnet wichtige Hinweise, die Sie unbedingt beachten sollten.

---

## 2 Einführung in POSIX

Dieses Kapitel wendet sich an alle Leser, die sich einen Überblick über POSIX verschaffen wollen. Es informiert Sie über die Rolle von POSIX im BS2000/OSD, das POSIX-Dateisystem, das Subsystem POSIX und über das Sicherheitskonzept von POSIX.

### 2.1 POSIX im BS2000/OSD

Mit POSIX setzt BS2000/OSD seine Strategie der Öffnung im Sinne der „Open Systems Direction“ konsequent fort.

Dieser Abschnitt gibt Ihnen einen allgemeinen Überblick über

- offene Systeme
- die Vorteile des POSIX-Standards
- die Bestandteile von POSIX
- die Hardware-Voraussetzungen für POSIX
- die Unterstützung von Terminals
- die BS2000-Softwarereprodukte, die an POSIX angepaßt wurden

## Neue Anforderungen an die Informationstechnologie

Viele Jahrzehnte waren die meisten Unternehmen stark hierarchisch strukturiert. In den letzten Jahren wurden flache Organisationsstrukturen eingeführt. „Lean Management“ sorgt für kürzere Wege und schnellere, flexiblere Entscheidungen.

Diese Entwicklung hat auch neue Anforderungen an die Informationstechnologie gestellt. Heute gehören leistungsfähige, kostengünstige PCs und Workstations zur Grundausstattung eines Arbeitsplatzes. Der zunehmende Wunsch der Anwender nach übergreifenden Lösungen erfordert es, daß diese PCs und Workstations mit den vorhandenen Hostsystemen zu einem optimalen Gesamtsystem kombiniert werden. Alle Systeme müssen miteinander kommunizieren können, um die gemeinsamen Ressourcen zu nutzen.

Die Vernetzung heterogener Systeme ist also eine Grundforderung an die Informationstechnologie. Die Wirtschaftlichkeit und Anwenderfreundlichkeit von PCs und Workstations müssen mit der hohen Rechnerleistung, Speicherkapazität, Verfügbarkeit, Datenkonsistenz und Sicherheit von Hostsystemen verbunden werden.

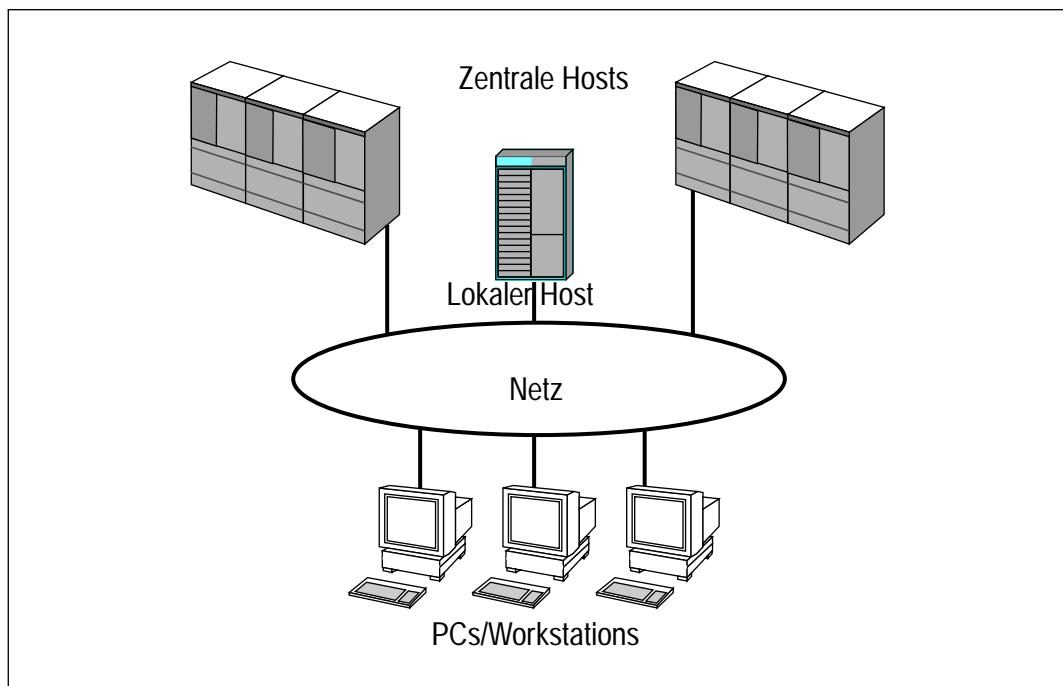


Bild 2: Vernetzung heterogener Systeme

Die Kommunikation zwischen heterogenen Systemen und die bessere Nutzung dieser Systeme ist nur möglich, wenn Standards für Betriebssystem-Schnittstellen festgelegt und eingehalten werden.

## Eine Welt offener Systeme

Im Auftrag des Institute of Electrical and Electronic Engineers (IEEE) wurden umfangreiche Standards für portierbare Betriebssystem-Schnittstellen entwickelt. Diese Standards wurden unter dem Begriff „POSIX“ zusammengefaßt. Durch POSIX werden proprietäre Systeme zu offenen Systemen. In offenen Systemen können Anwendungen über Systemgrenzen hinweg übertragen werden (Portabilität) und mit anderen Anwendungen zusammenarbeiten (Interoperabilität).

Siemens Nixdorf ist seit Jahren Mitglied in den Konsortien und Standardisierungsgremien, die sich um die Gestaltung internationaler offener Standards bemühen. So ist Siemens Nixdorf z.B. Gründungsmitglied von X/OPEN und OSF (**Open Software Foundation**), die mittlerweile in „The Open Group“ übergegangen sind.

## Offenes BS2000 - BS2000/OSD

Die Siemens Nixdorf Informationssysteme AG hat die Offenheit ihres Produkt- und Systemangebots als Schwerpunkt der strategischen Geschäftsausrichtung in der „Open Systems Direction“ (OSD) festgelegt. Deshalb wird auch das Betriebssystem BS2000 verstärkt auf eine offene Systemwelt ausgerichtet. Diese Ausrichtung macht auch der neue Name „BS2000/OSD“ sichtbar.

Die wichtigsten Elemente von Offenheit sind garantierte Systemschnittstellen, die Siemens Nixdorf in seiner strategischen Schnittstellenplattform SIA (System Interfaces for Applications) festgeschrieben hat. SIA stützt sich auf vorhandene Standards für Application Programming Interfaces (API), die auf den Plattformen BS2000/OSD, Reliant UNIX, Windows und Windows NT einheitlich sind. Wo Standards noch ausstehen, werden von Siemens Nixdorf spezifische APIs angeboten und langfristig garantiert.

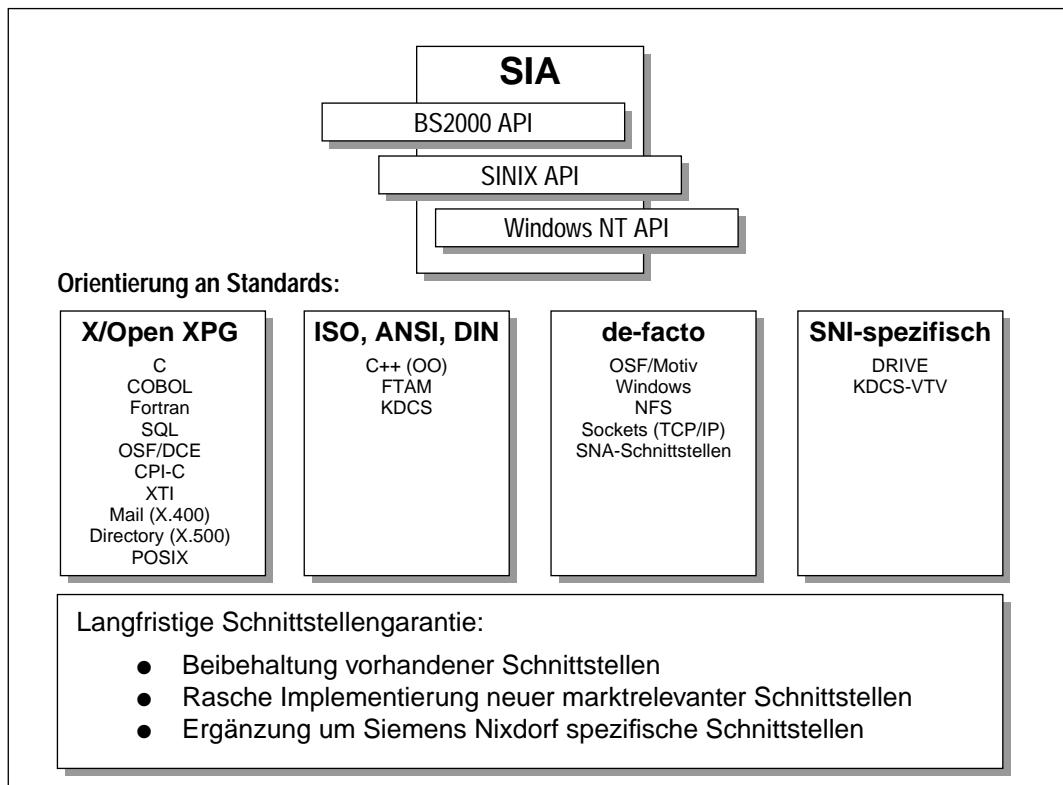


Bild 3: SIA-Schnittstellenkonzept

Die unterschiedlichen Systeme von verschiedenen Herstellern sollen über offene Standardschnittstellen optimal miteinander kommunizieren können. Deshalb konzentriert sich die Weiterentwicklung des Betriebssystems BS2000 unter anderem darauf, die Integrationsfähigkeit in offene Systemumgebungen mit unterschiedlichen Systemplattformen zu erweitern. Dadurch können Standardanwendungen auf unterschiedlichen Systemplattformen ablaufen.

BS2000/OSD schafft weltweite Verbindungen auf Basis der international in Netzen verwendeten Normen. Die Siemens-Nixdorf-Netze bieten die Anbindung von Workstations aller gängigen Betriebssysteme (UNIX/SINIX, Windows 3.1, Windows 95, Windows NT, ...) an BS2000-Systeme. Sie ermöglichen verteilte Transaktionsverarbeitung im homogenen und heterogenen Rechnerverbund. BS2000-Systeme übernehmen - zusätzlich zur traditionellen Mainframe-Funktion - immer öfter die Rolle eines offenen Servers in Client-Server-Konfigurationen.

BS2000/OSD ist eine kompatible Erweiterung des Betriebssystems BS2000. Deshalb laufen bestehende BS2000-Anwendungen in BS2000/OSD wie bisher ab; alle Dienste des BS2000 sind nach wie vor verfügbar.

BS2000/OSD stellt neue Kommandos zur Verfügung, so daß BS2000-Dateien für den Zugriff von POSIX-Anwendungen kopiert werden können. Benutzer (oder auch Programme, die fertige Prozeduren oder Shell-Scripts aufrufen) können BS2000-Dateien in ein hierarchisches POSIX-Dateisystem kopieren.

Viele Standard-Softwarepakete sind bereits heute auf BS2000/OSD verfügbar. So das relationale Datenbanksystem ORACLE und die betriebswirtschaftliche Anwendung R/2. Bei Bedarf lassen sich weitere Softwarepakete aus der offenen Welt kostengünstig portieren.

## Offenheit durch Client-Server-Architekturen

In Client-Server-Architekturen werden verschiedene Rechnerwelten systemübergreifend integriert. Dezentrale, intelligente Rechner werden mit zentralen Mainframes verbunden. Dadurch ist eine verteilte Verarbeitung möglich.

Innerhalb dieses homogenen Ganzen stellen die Server verschiedene Dienste zur Verfügung, die von den Clients genutzt werden. Server-Funktionen werden überwiegend von Mainframe- und UNIX-Systemen ausgeübt. Client-Funktionen haben vor allem PCs und Workstations, aber auch UNIX-Systeme. Client- und Server-Systeme können beliebig kombiniert werden.

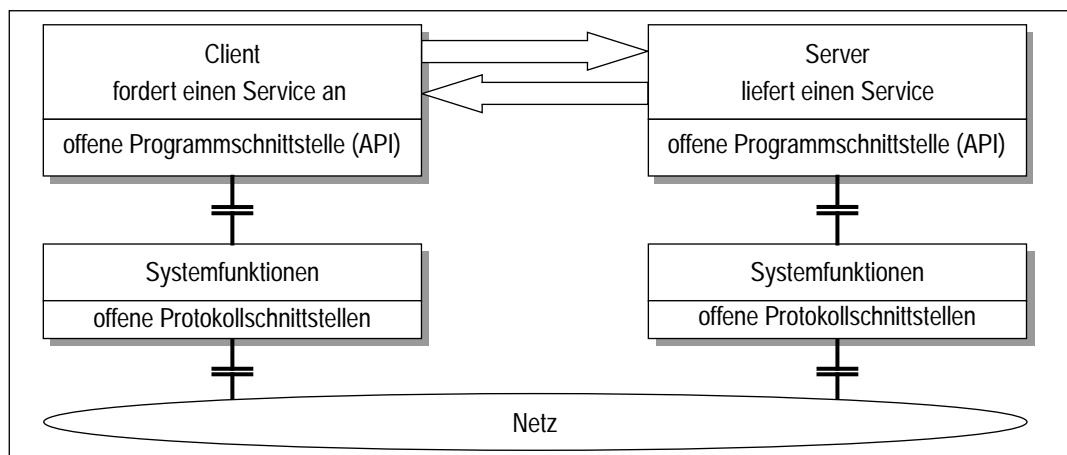


Bild 4: Rollenverteilung in einer Client-Server-Architektur

In einer flexiblen Architektur kann derselbe Rechner für bestimmte Dienste als Client und für andere Dienste als Server eingesetzt werden. So lassen sich die Stärken der verschiedenen Rechner optimal nutzen:

- Auf PCs laufen bevorzugt Standardanwendungen unter MS Windows ab, besonders im Bereich der Textverarbeitung, Tabellenkalkulation und der Geschäftsgrafik.
- Bei den Workstations stehen Anwendungen im Vordergrund, die eine hohe Grafikleistung erbringen, wie z.B. CAD (Computer Aided Design).
- BS2000-Mainframes eignen sich besonders als unternehmensweite Server durch ihre sehr große Rechnerleistung, den Einsatz von großen Massenspeichern, die hohe Sicherheit und den hohen Automatisierungsgrad der Administration.

BS2000-Server arbeiten mit anderen Servern im Netz zusammen, unabhängig davon, ob es sich um BS2000- oder um UNIX-Server handelt. So können zum Beispiel abteilungsinterne Daten von verschiedenen UNIX-Servern verwaltet werden, während für übergreifende Daten ein zentraler BS2000-Server mit Hochleistungsperipherie zuständig ist.

Für den Anwender haben Client-Server-Architekturen mehrere Vorteile:

- Die Flexibilität in der Ablauforganisation wird erhöht.
- Informationen sind leicht und überall verfügbar.
- Unter den verschiedenen Systemen wird eine optimale Lastverteilung erreicht.
- Das Rechnernetz kann der jeweiligen Betriebsgröße genau angepaßt werden.

## BS2000/OSD bringt die UNIX- und die BS2000-Welt zusammen

Workstation-Benutzer können über die POSIX-Schnittstelle die Ressourcen und die Leistungsstärke des BS2000 nutzen, ohne die BS2000-propriären Schnittstellen kennen zu müssen. Im Verbund mit BS2000/OSD wird der Plattspeicher einer Workstation um den Host-Plattspeicher erweitert, damit auch der Workstation-Benutzer große Datenmengen verarbeiten kann. Beispielsweise kann ein Entwickler seine Anwendungen auf einer Workstation entwickeln und anschließend seine Programme im BS2000 übersetzen, testen, korrigieren und ablaufen lassen.

Die UNIX- und die BS2000-Welt können unabhängig voneinander existieren, wobei sie sich aber dieselben Prozeß- und Speicherressourcen teilen. Die Benutzer können sich das Beste aus beiden Welten aussuchen: Die Standardschnittstelle und Portabilität des UNIX-Teils und eine Vielzahl von Diensten des BS2000-Teils.

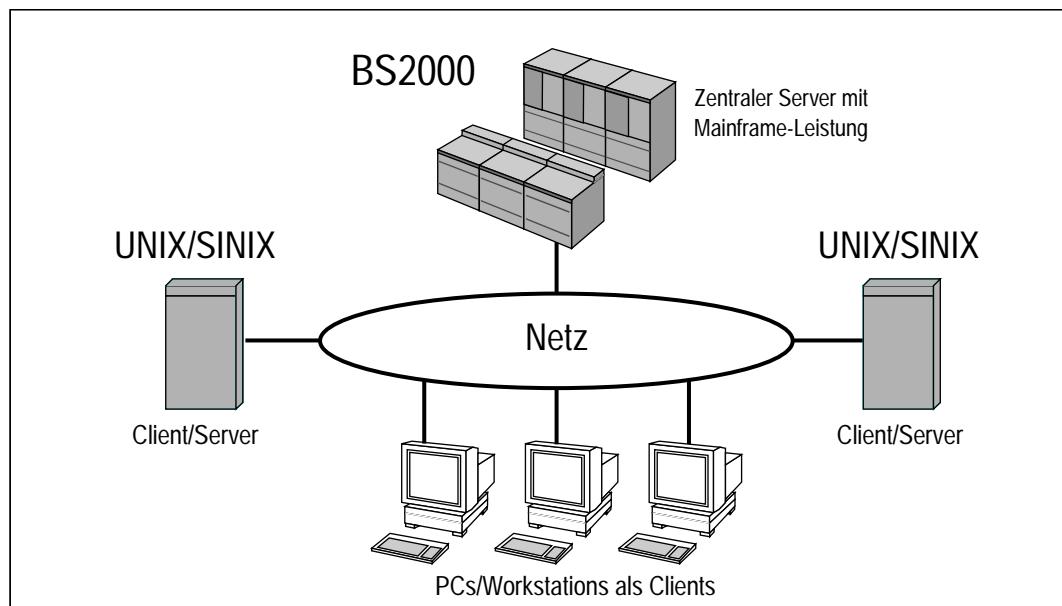


Bild 5: Stellung des BS2000 im Rechnerverbund

## Vorteile des POSIX-Standards

Der POSIX-Standard bietet Ihnen folgende Vorteile:

- Portabilität von Anwendungsprogrammen
- Interoperabilität von Anwendungsprogrammen
- Arbeiten mit hierarchischen Dateisystemen
- BS2000/OSD als Server
- Verteilte Datenhaltung
- Verteilte Verarbeitung
- Gemeinsame Entwicklungstools

Im folgenden sind diese Vorteile näher erläutert.

### Portabilität von Anwendungsprogrammen

Anwendungsprogramme, die gemäß den POSIX-Schnittstellen geschrieben sind, können auf allen XPG4-konformen Betriebssystemen und Hardware-Plattformen ablaufen. Portable Anwendungsprogramme können im BS2000 ebenso problemlos ablaufen wie zum Beispiel auf einer UNIX-Plattform.

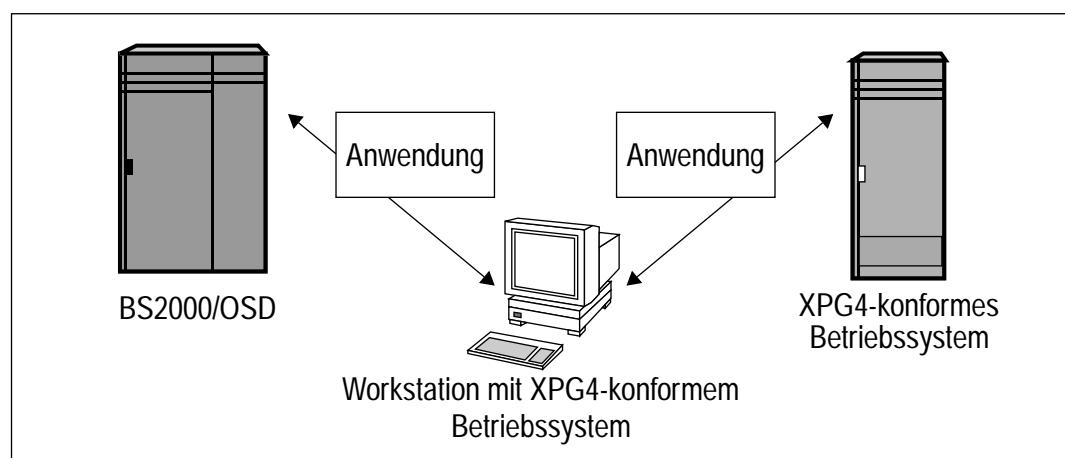


Bild 6: Portieren einer Anwendung auf ein beliebiges XPG4-konformes System

Daten und Dateien von Anwendungsprogrammen im ASCII-Code müssen vor dem Einsatz in POSIX in EBCDIC konvertiert werden (siehe Abschnitt „Kopieren und Konvertieren von Dateien“ auf Seite 27).

## Interoperabilität von Anwendungsprogrammen

Anwendungsprogramme, die unter verschiedenen XPG4-konformen Betriebssystemen ablaufen, können Daten untereinander austauschen, falls die Dateiformate übereinstimmen (siehe Abschnitt „Kopieren und Konvertieren von Dateien“ auf Seite 27).

## Bereitstellung hierarchischer Dateisysteme

Mit dem POSIX-Dateisystem wird das BS2000 um ein hierarchisch strukturiertes Dateisystem erweitert. Ein POSIX-Dateisystem ist eine Behälterdatei (Container) im BS2000 mit der Struktur eines UNIX-Dateisystems (UFS). Das POSIX-Dateisystem besteht aus Dateien (POSIX-Dateien) und Dateiverzeichnissen (näheres siehe Abschnitt „POSIX-Dateisystem“ auf Seite 24). POSIX-Benutzer können POSIX-Dateien erzeugen und bearbeiten.

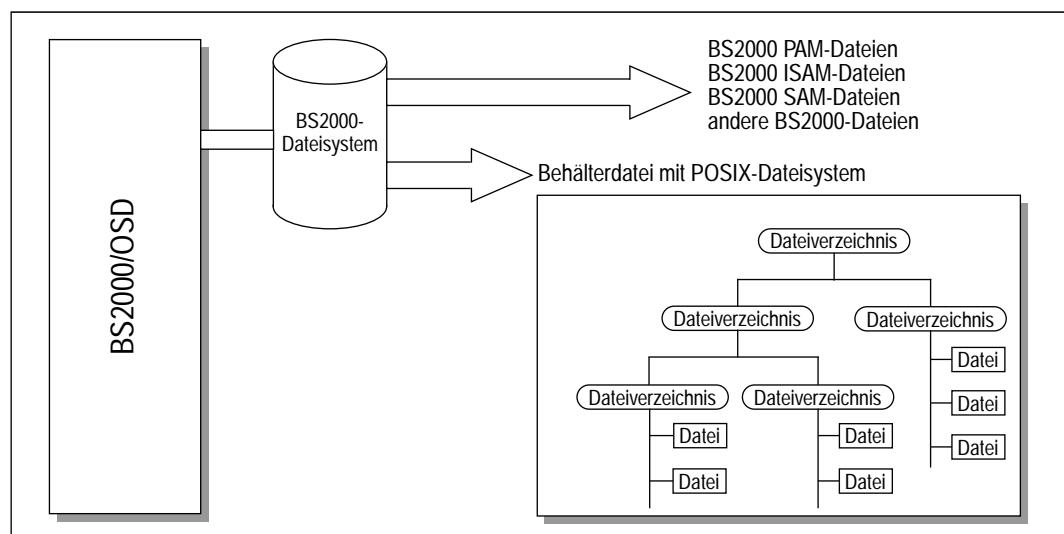


Bild 7: Unterstützung von POSIX-Dateisystemen durch BS2000/OSD

Mit dem Softwareprodukt NFS (Network File System) lassen sich lokale POSIX-Dateisysteme in ferne Rechner einhängen und ferne UFS-Dateisysteme in ein lokales POSIX-Dateisystem.

## BS2000/OSD als Server

BS2000/OSD kann als reiner Datenserver eingesetzt werden. Dabei befinden sich die Daten (Datenbanken und Dateien) auf einem BS2000-Rechner. Die Anwendungen sind auf einem anderen Rechner abgelegt. Dies ist bei einer geringen Zahl von Datenzugriffen pro Transaktion sinnvoll.

Beim Einsatz als Server für Anwendungen und Daten befinden sich die Anwendungen und die Daten auf demselben BS2000-Rechner. Dies ist dann vorteilhaft, wenn pro Transaktion häufig auf die Daten zugegriffen wird.

Als File-Server bietet BS2000/OSD die Kapazität, Zugriffsgeschwindigkeit und Zugriffs-sicherheit seiner Speichersubsysteme.

Als Backup-Server kann BS2000/OSD Datenbestände aus dem Netz auf seinen Speicher-medien hinterlegen und die dort verfügbaren Sicherungsmechanismen von HSMS nutzen (siehe Handbuch „HSMS / HSMS-SV“ [22]).

Als Print-Server stellt BS2000/OSD seine Drucker über ein verteiltes Spool- und Druck-system zur Verfügung (siehe Handbuch „SPOOL“ [31] und [32]). Dadurch können die Benutzer von UNIX-Arbeitsplätzen ihre Druckaufträge an BS2000-Druckern schnell und kostengünstig ausdrucken.

## POSIX und das World Wide Web (WWW)

Ab BS2000/OSD-BC V2.0 steht mit dem Produkt *WebServe* eine WWW-Serversoftware für den Anschluß von BS2000-Hosts ans Internet zur Verfügung (siehe Handbuch „*WebServe*“ [36] oder im WWW unter [http://www.siemensnixdorf.com/servers/bs2\\_man/man\\_de.htm](http://www.siemensnixdorf.com/servers/bs2_man/man_de.htm)).

Basis für *WebServe* ist der W3C *httpd* (Hypertext-Transferprotokoll- Dämon), der eine kostenlose Ergänzung zu POSIX-BC ist. Damit kann das BS2000/OSD mit seinem inte-grierten Subsystem POSIX sämtliche HTML-Dokumente inklusive Grafiken, Videoclips und Sounddateien auf einem BS2000/OSD-Server halten und anbieten.

Auch BS2000/OSD-Anwendungen können sich mit ihren Daten im Internet präsentieren. Über ein Common Gateway Interface (CGI), das durch den WWW-Server im BS2000/OSD verfügbar ist, wird der direkte Zugriff auf die Anwendungen realisiert. Diese Anwendungen können mit oder ohne Transaktionssteuerung durch UTM ablaufen und auf beliebige BS2000/OSD-Datenhaltungen zugreifen. Informationen zum CGI finden Sie im WWW unter <http://hoohoo.ncsa.uiuc.edu/cgi> .

Die Sicherheitsfunktionen von BS2000/OSD gewährleisten, daß nicht alle Daten jedem frei zugänglich sind.

### BS2000/OSD-Anwendungen ins WWW bringen

Bestehende BS2000/OSD-Anwendungen können mit geringem Aufwand WWW-fähig gemacht werden. Dabei werden mit dem Produkt *WebTransactions* alphanumerische Oberflächen (Masken) in HTML-Formate umgesetzt und den WWW-Browsern zur Ausgabe übergeben. Weiterführende Informationen dazu finden Sie im Handbuch "WebTransactions" [38].

Die Zahl der angebotenen Host-Anbindungen wird ständig erweitert. Spezialanbindungen werden mit der nötigen Einbindungstechnik momentan als Projektleistung von SNI angeboten (siehe dazu auch [37]).

Die WWW-Einbindung mit *WebTransactions* ist einfach; die Anwendungen können entweder mit ihrer bekannten Oberfläche oder optisch aufgewertet im WWW gezeigt und bedient werden.

### Verteilte Datenhaltung

Bei der verteilten Datenhaltung können Sie sowohl mit lokalen als auch mit fernen Daten arbeiten. Dadurch können Sie Datenbestände an die kostengünstigste Stelle in einem Rechnernetz legen.

Sie können von einer Workstation auf BS2000-Dateien zugreifen, nachdem die BS2000-Dateien in ein POSIX-Dateisystem kopiert wurden. Sie können aber auch BS2000-Dateien in ein POSIX-Dateisystem kopieren und auf eine Workstation laden.

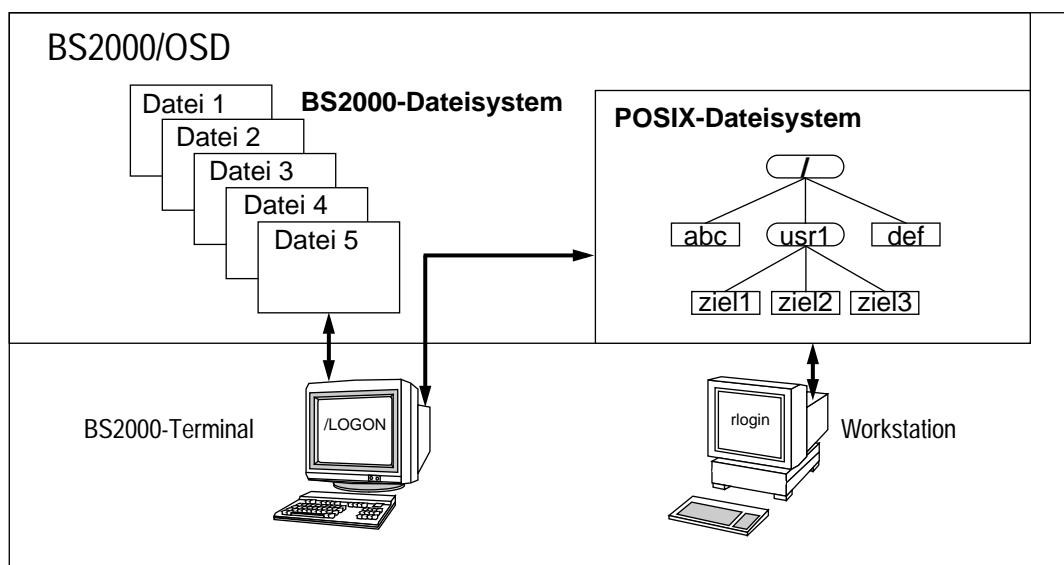


Bild 8: Verteilte Datenhaltung

## Verteilte Verarbeitung

Mit der verteilten Verarbeitung können Sie die Rechnerkapazitäten optimal nutzen. Bei dieser Form der Client-Server-Architektur findet die Verarbeitung am jeweils geeigneten Ort statt. Beispielsweise können Eingabeprüfungen und Kalkulationen auf einer Workstation ablaufen, während Datenbankzugriffe und rechenintensive Auswertungen auf einem BS2000-Rechner laufen.

Für die verteilte Verarbeitung steht im BS2000/OSD das Softwareprodukt DCE (**Distributed Computing Environment**) der Open Software Foundation zur Verfügung. DCE setzt im BS2000 auf POSIX auf. Es benutzt das POSIX-Dateisystem und die UNIX-ähnliche Umgebung, die POSIX bietet.

DCE stellt Programmschnittstellen und Dienste zur Verfügung, die die Entwicklung und den Ablauf von verteilten Anwendungen unterstützen, wie z.B. Threads, RPC, Directory-, Time- und Security-Service.

Nähere Informationen finden Sie in den Handbüchern zu „DCE“ (siehe [8] bis [11] im Literaturverzeichnis).

## Gemeinsame Entwicklungstools

UNIX-Entwicklungstools können ohne großen Aufwand nach POSIX portiert werden. Damit lassen sich Anwendungsprogramme erstellen, die auf UNIX-Systemen und auf POSIX ablaufen können.

## Bestandteile von POSIX

POSIX setzt sich aus folgenden Bestandteilen zusammen:

- **Subsystem POSIX**  
Es wird mit BS2000/OSD-BC ab V2.0 geliefert und beinhaltet einen ins BS2000/OSD portierten SINIX-Systemkern.
- **POSIX-Shell**  
Sie stellt mit den Shell-Kommandos entsprechend dem XPG4-Standard die Verbindung zwischen dem Systemkern und dem BS2000/OSD her.
- **C-Bibliothek**  
Sie stellt die Programmschnittstellen entsprechend dem XPG4-Standard bereit und ist Bestandteil des Softwareprodukts CRTE.
- **POSIX-Sockets und XTI**  
Die Programmschnittstellen für die Transportsystem- und Kommunikationsdienste werden mit POSIX-Sockets und XTI (X/Open Transport Interface) bereitgestellt und sind Bestandteil von BS2000/OSD-BC ab V2.0.
- **WebServe**  
Ab BS2000/OSD-BC V2.0 steht mit dem Produkt *WebServe* eine WWW-Serversoftware für den Anschluß von BS2000-Hosts ans Internet zur Verfügung.

## Hardware-Voraussetzungen für POSIX

POSIX kann auf allen Zentraleinheiten ablaufen, auf denen BS2000/OSD-BC ab V2.0 ablauffähig ist. Die Versionsabhängigkeiten zwischen POSIX-BC und BS2000/OSD-BC entnehmen Sie bitte der Freigabemitteilung zu POSIX-BC.

## Unterstützung von Terminals

POSIX unterstützt neben den im BS2000 verwendeten Blockterminals auch die an UNIX-Systemen verwendeten Zeichenterminals. Diese Terminals sind an UNIX-Mehrplatzsystemen angeschlossen und werden von POSIX über Netze bedient. Beim Zugriff auf POSIX über eine Workstation wird ein Zeichenterminal emuliert. Bei SINIX-Workstations ist das z.B. ein Terminal vom Typ 97801.

Block- und Zeichenterminals unterscheiden sich durch ihre Funktionsweise und die Verbindung zum BS2000:

- Blockterminals sind direkt mit dem BS2000 und POSIX verbunden. Von Blockterminals aus können BS2000- und POSIX-Kommandos eingegeben werden, wobei aber die Eingabe von POSIX-Kommandos geringfügigen Einschränkungen unterliegt (siehe Handbuch „POSIX-Kommandos“ [2]).

Bei Blockterminals ist keine zeichenweise Verarbeitung möglich. Blockterminals übergeben den gesamten am Bildschirm eingegebenen Text als Datenblock an den BS2000-Rechner. Kontrollfunktionen werden im Gerät selbst durchgeführt.

- Zeichenterminals sind normalerweise an UNIX-Rechnern angeschlossen, die mit dem BS2000-Host über TCP/IP-LAN verbunden sein müssen. Mit der *telnet*-Anwendung (siehe Seite 49) kann sich ein PC oder eine Workstation wie ein Zeichenterminal bei POSIX anmelden (ohne den Umweg über eine Terminal-emulation und *rlogin* an einem UNIX-Host).

Bei Zeichenterminals wird jedes eingegebene Zeichen sofort an den UNIX-Rechner übertragen und von dort als Antwort auf die Eingabe an den Bildschirm übergeben und abgebildet. Kontrollfunktionen wie Schreibmarken-Bewegung, Groß-/Kleinschreibung oder Pufferung der Übertragung werden von dem Rechner durchgeführt, an dem das Terminal angeschlossen ist.

Zeichenterminals werden in POSIX wie Dateien behandelt. Sie besitzen einen eindeutigen Namen. Von ihnen kann gelesen und auf sie kann geschrieben werden. Dazu werden die gleichen Funktionen wie beim Dateizugriff benutzt.

Bildschirm-orientierte Anwendungen - wie z.B. der vi-Editor in UNIX-Systemen - erfordern zeichenorientierte Operationen. Deshalb können sie nur ablaufen, wenn sie an einem Zeichenterminal gestartet werden.

Manche Eingaben sind terminalabhängig, d.h. sie unterscheiden sich bei Block- und Zeichenterminals:

Blockterminal	Zeichenterminal
@@d	[END]
@@c	[DEL]
@@/	[CTRL] [V]
[EM] [DUE]	[↓]
-	[CTRL] [S] / [CTRL] [Q] / [CTRL] [Z] ...

Eine Aufteilung des Bildschirms existiert im BS2000 nicht. Die Bildschirmdarstellung erfolgt immer von oben nach unten. Eingaben und Ausgaben erfolgen jeweils in der untersten aktiven Zeile. Wenn der Bildschirm voll ist, wird der Inhalt jeweils um eine Zeile nach oben geschoben. Die oberste Zeile geht dadurch verloren. Ein Zugriff auf vorhergehende Zeilen ist nicht mehr möglich.

## An POSIX angepaßte BS2000-Softwareprodukte

POSIX-BC ist ein Subsystem von BS2000/OSD. Verschiedene BS2000-Softwareprodukte wurden an die POSIX-Schnittstellen angepaßt oder auf POSIX-Basis ins BS2000 portiert:

- Binder-Lader-System ab BS2000/OSD-BC V2.0
- C/C++-Compiler ab V2.2
- CRTE ab V2.0A
- EDT ab V16.5
- File-Transfer-Produkte (FT-BS2000, FTOS-BS2000, *openFT*)
- HSMS / HSMS-SV ab V2.0B
- NFS ab V1.2
- SDF-A ab V4.0
- SECOS ab V2.1
- SOCKETS/XTI (POSIX-SOCKETS) ab V1.1
- SPOOL ab V3.0
- TLI (POSIX-NSL) ab V1.0
- AID ab V2.2
- SORT ab V7.5
- TCP-IP-AP ab V2.0
- TCP-IP-SV
- DCE (BS2000)
- TransView SNMP-Basic-Agent BS2000

Näheres dazu finden Sie im Kapitel „BS2000-Softwareprodukte im Umfeld von POSIX“ ab Seite 59.

## 2.2 POSIX-Dateisystem

Ein POSIX-Dateisystem ist eine Behälterdatei (Container) im BS2000 mit der Struktur eines UNIX-Dateisystems (UFS). Es kann wie in UNIX aus mehreren Dateisystemen bestehen. Es ist hierarchisch aufgebaut und besteht aus Dateiverzeichnissen und Dateien (POSIX-Dateien).

An der Spitze der Hierarchie steht das Dateiverzeichnis *root*, das durch einen Schrägstrich (/) gekennzeichnet ist. Von hier aus setzt sich die Verzeichnisstruktur weiter nach unten fort. Von Dateiverzeichnissen aus kann in weitere Dateiverzeichnisse oder in Dateien verzweigt werden. Eine Datei ist der tiefste Verzweigungspunkt. Von einer Datei aus ist keine Verzweigung mehr möglich.

Einschränkungen gibt es weder für die Anzahl der Verzeichnisebenen noch für die Anzahl der Dateiverzeichnisse und Dateien auf einer Ebene. Deshalb lässt sich ein POSIX-Dateisystem sehr gut strukturieren und organisieren.

Dateiverzeichnisse werden auch als Knotenpunkte eines POSIX-Dateisystems bezeichnet, in denen Namen von Dateien oder weiteren Dateiverzeichnissen stehen. Die Namen für die Dateiverzeichnisse und Dateien kann der Benutzer vergeben, wobei bestimmte Konventionen einzuhalten sind.

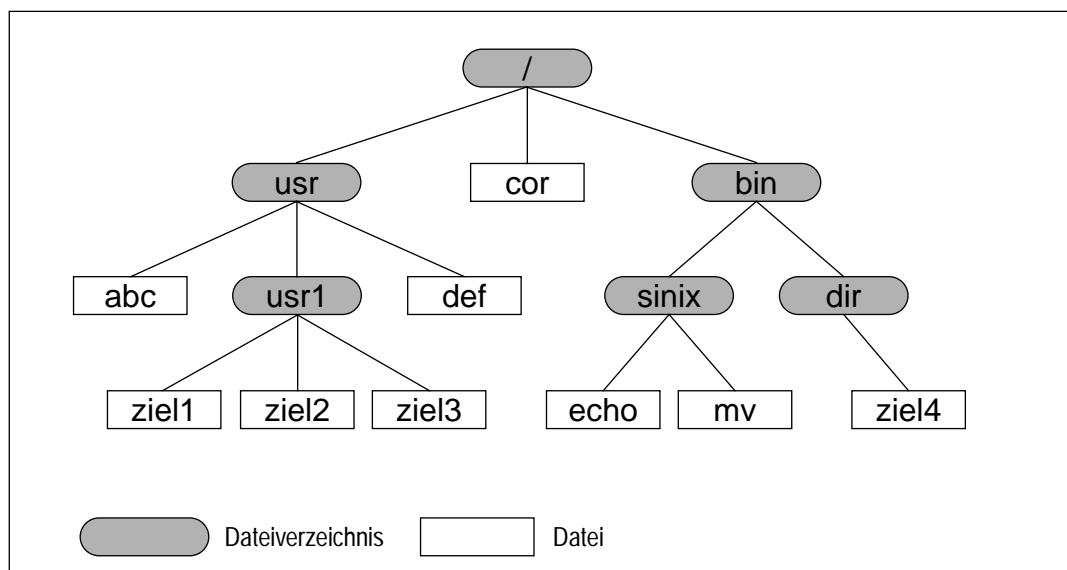


Bild 9: Hierarchische Struktur eines POSIX-Dateisystems

## Vorteile eines hierarchischen Dateisystems

Ein hierarchisch strukturiertes Dateisystem wie das POSIX-Dateisystem bietet Ihnen mehrere Vorteile:

- Sie können ihren Datenbestand besser strukturieren.
- Sie können mit jeder Datei oder jedem Dateiverzeichnis des gesamten Dateisystems arbeiten, wenn Sie für die entsprechende Datei oder das entsprechende Dateiverzeichnis eine Zugriffsberechtigung besitzen (siehe Abschnitt „Zugriffsschutz für Dateien und Dateiverzeichnisse“ auf Seite 41).
- Eine Datei lässt sich leicht vom aktuellen Dateiverzeichnis in ein anderes Dateiverzeichnis übertragen. Dazu gibt es drei Möglichkeiten:
  - Sie können mit dem POSIX-Kommando *cp* (copy) eine Datei physisch in ein anderes Dateiverzeichnis kopieren; die Datei ist dann physisch mehrfach vorhanden.
  - Sie können aber auch nur den Namen einer Datei mit dem POSIX-Kommando *ln* (link) in ein anderes Dateiverzeichnis übertragen. Auf eine solche Datei bestehen dann mehrere Verweise; die Datei ist physisch aber nur einmal vorhanden.
  - Sie können mit dem POSIX-Kommando *mv* (move) eine Datei umbenennen oder an einen anderen Ort im Dateibaum versetzen. *mv* erzeugt innerhalb eines Dateisystems keine physische Kopie der versetzten oder umbenannten Datei, sondern ändert nur die Einträge im jeweils übergeordneten Dateiverzeichnis.

Die POSIX-Kommandos *cp*, *ln* und *mv* sind im Handbuch „POSIX-Kommandos“ [2] ausführlich beschrieben.

- Sie können Ihre Dateien in ein oder mehrere Dateiverzeichnisse schreiben. Dadurch lassen sich Dateien übersichtlich und zusammenhängend organisieren.
- In einem Dateisystem dürfen mehrere Dateien mit demselben Namen vorhanden sein. Die Dateien müssen aber in unterschiedlichen Dateiverzeichnissen abgelegt sein.

## Ablage von POSIX-Dateisystemen in Behälterdateien

POSIX-Dateisysteme werden im BS2000 in sog. Behälterdateien abgelegt; dies entspricht der im UNIX üblichen Ablage von Dateisystemen in Partitions. Behälterdateien sind BS2000-PAM-Dateien, die sich auf einem Pubset befinden. Behälterdateien dürfen nicht auf Shared Public Volume Sets (SPVS) und auf Privatplatten abgelegt werden. Behälterdateien und andere BS2000-Dateien dürfen auf dem gleichen Pubset liegen.

Eine Behälterdatei darf nicht mit der Zugriffsmethode UPAM bearbeitet werden.

Aus der Sicht des Subsystems POSIX stellt eine Behälterdatei ein Dateisystem dar, das vom portierten SINIX-Systemkern verwaltet wird.

POSIX-Verwalter und BS2000-Systemverwalter mit Root-Berechtigung können Behälterdateien beim Einrichten neuer POSIX-Dateisysteme mit dem POSIX-Installationsprogramm anlegen (siehe Seite 91). Dabei wird auch die Größe der Behälterdatei und damit die des POSIX-Dateisystems festgelegt. Die Größe kann nachträglich nicht mehr geändert werden.

Um für einen Benutzer den Speicherplatz zu begrenzen, kann für diesen Benutzer ein eigenes POSIX-Dateisystem mit der entsprechenden Größe eingerichtet werden. Dadurch wird der vorhandene Speicherplatz im BS2000 wirtschaftlicher genutzt.

Aus Performancegründen sollten die Behälterdateien von großen POSIX-Dateisystemen, die häufig benutzt werden, nicht auf demselben Pubset liegen, auf dem sich die Behälterdatei des Root-Dateisystems befindet.

## Vorteile durch das Anlegen mehrerer POSIX-Dateisysteme

Das Anlegen mehrerer POSIX-Dateisysteme bringt folgende Vorteile:

- Größere Datensicherheit

Bei der Zerstörung eines POSIX-Dateisystems bleiben die übrigen POSIX-Dateisysteme erhalten.

Einzelne POSIX-Dateisysteme können für eine Sicherung ausgewählt werden. Unveränderte POSIX-Dateisysteme können von einer Sicherung ausgenommen werden.

- Größerer Datenschutz

Nur aktuell benötigte POSIX-Dateisysteme werden eingehängt und damit dem Benutzer verfügbar gemacht.

- Bessere Übersichtlichkeit und Strukturierung

Ein POSIX-Dateisystem kann speziell für einen Benutzer oder für ein Projekt angelegt werden.

## Konventionen für Namen von POSIX-Dateien und Dateiverzeichnissen

Jede Datei und jedes Dateiverzeichnis in einem POSIX-Dateisystem hat einen eindeutigen Pfadnamen. Der Pfadname gibt die Position einer Datei oder eines Dateiverzeichnisses innerhalb eines POSIX-Dateisystems an und zeigt, wie darauf zugegriffen werden kann. Der Pfadname besteht aus den Namen aller darüberliegenden Dateiverzeichnisse, ausgehend vom Dateiverzeichnis *root*, und dem eigentlichen Namen der Datei oder des Dateiverzeichnisses. Die Namen der Dateiverzeichnisse werden jeweils durch einen Schrägstrich (Slash) voneinander getrennt. Wenn man vom POSIX-Dateisystem in Bild 9 ausgeht, dann hat z.B. der Pfad vom Dateiverzeichnis *root* zur Datei *echo* folgenden Namen: */bin/sinix/echo*

Wenn Sie eine Datei oder ein Dateiverzeichnis ohne Pfadangabe einrichten, wird der Name automatisch immer in dem Dateiverzeichnis eingetragen, in dem Sie sich gerade befinden.

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.

## Kopieren und Konvertieren von Dateien

POSIX-Dateien enthalten keine Datensätze, sondern sie sind byte-orientiert.

BS2000-Dateien dagegen enthalten satzorientierte und/oder PAM-Block-orientierte Daten.

POSIX behandelt Dateien standardmäßig im EBCDIC-Format, SINIX/UNIX, MS-DOS und Windows im ASCII-Format. Im POSIX-Dateisystem abgelegte ASCII-Dateien können in der POSIX-Shell nur bearbeitet werden, wenn sie vorher konvertiert wurden.

Damit Dateien der beiden Formate wechselseitig benutzt werden können, stehen Kopier- und Konvertierungsroutinen zur Verfügung. Dabei wird vom Zeichensatz EBCDIC.DF.03 in den ASCII-ISO-7-Bit-Code konvertiert und umgekehrt.

Die Konvertierung ist nur für Textdateien sinnvoll.

## Automatische Konvertierung

Mit der Umgebungsvariable *IO\_CONVERSION* wird gesteuert, ob Dateien beim Zugriff mit POSIX-Kommandos (z.B. *awk*, *cat*, *grep...*) auf montierte ASCII-Dateisysteme automatisch konvertiert werden. Standardmäßig ist die Umgebungsvariable *IO\_CONVERSION* mit dem Wert „NO“ belegt, d.h. es erfolgt keine automatische Konvertierung. Die automatische Konvertierung wird mit folgendem Kommando eingeschaltet:

```
export IO_CONVERSION=YES
```

Falls die automatische Konvertierung für einen POSIX-Benutzer bereits beim Starten der POSIX-Shell voreingestellt sein soll, muß dieses *export*-Kommando in die Datei *.profile* im HOME-Verzeichnis dieses Benutzers eingetragen werden.

**i** Bei Verwendung der folgenden Tools darf die automatische Konvertierung nicht eingeschaltet sein, da diese Tools selbst konvertieren:  
*dd, iconv, edt mit Schalter -k, bs2cp mit Schalter -k.*

## Behandlung von Archiven/Bibliotheken:

ar konvertiert nicht automatisch, da ar-Bibliotheken oft binäre Daten enthalten.

*pax* und *tar* konvertieren automatisch. Ein *pax*- oder *tar*-Archiv darf jedoch nicht mit *cp* kopiert werden, wenn die automatische Konvertierung eingeschaltet ist.

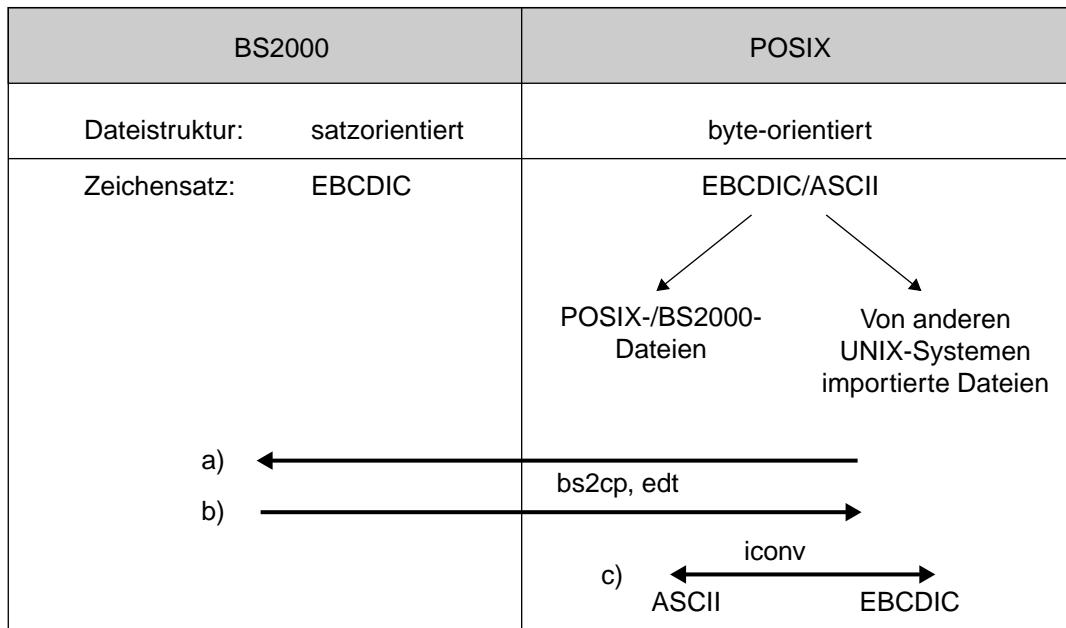


Bild 10: Austausch und Konvertierung von Dateien

- a) Dateien von POSIX nach BS2000 übertragen (aus Sicht der POSIX-Shell):

Mit dem POSIX-Kommando *bs2cp* oder *edt* übertragen Sie Dateien von POSIX nach BS2000. Sie müssen nicht die Option *-k* angeben, wenn in beiden Dateisystemen die Dateien im EBCDIC-Zeichensatz vorliegen.

Zusätzlich können Sie für die BS2000-Datei noch Dateiattribute bestimmen. Dazu müssen Sie vor dem „Kopier“kommando *bs2cp* mit dem POSIX-Kommando *bs2file* die BS2000-Dateiattribute festlegen. *bs2file* wird auf das BS2000-Kommando SET-FILE-ATTRIBUTES abgebildet.

- b) Dateien von BS2000 nach POSIX übertragen (aus Sicht der POSIX-Shell):

Mit dem POSIX-Kommando *bs2cp* oder *edt* übertragen Sie Dateien von BS2000 nach POSIX. Sie müssen nicht die Option *-k* angeben, wenn in beiden Dateisystemen die Dateien im EBCDIC-Zeichensatz vorliegen.

Abhängig von der Art der BS2000-Datei (SAM, ISAM) ist folgendes zu berücksichtigen:

- Bei SAM-Dateien können Sie wählen, ob die Datei als Textdatei, als Binärdatei oder als binäre Textdatei im POSIX-Dateisystem hinterlegt wird. Dazu müssen Sie vor dem Kopierkommando *bs2cp* noch die Bearbeitungsart der Datei mit dem POSIX-Kommando *ftyp* festlegen.
  - ISAM-Dateien werden generell als Textdateien im POSIX-Dateisystem abgelegt.
- c) Zur Konvertierung innerhalb eines POSIX-Dateisystems dient das POSIX-Kommando *iconv*. Es werden die Datei-Inhalte konvertiert.

## Zugriff auf POSIX-Dateisysteme im BS2000

Beim Zugriff auf POSIX-Dateisysteme im BS2000 in **EBCDIC-Code** ist folgendes zu beachten:

- Beim Zugriff aus dem BS2000 sind keine Maßnahmen erforderlich.
- Beim Zugriff aus SINIX müssen die Dateien erst konvertiert werden, z.B. mit dem POSIX-Kommando *iconv* (siehe Handbuch „POSIX-Kommandos“ [2]).

Beim Zugriff auf POSIX-Dateisysteme im BS2000 in **ASCII-Code** ist folgendes zu beachten:

- Beim Zugriff aus dem BS2000:

CRTE bietet ab V2.0A eine *automatische* Konvertierung in EBCDIC-Code an, wobei aber folgende Einschränkungen gelten:

- Das zugehörige Dateisystem darf nicht als „von POSIX erzeugt“ markiert sein (siehe „Angaben zum POSIX-Dateisystem:“ auf Seite 90).
- Die Datei muß mit dem Aufruf *fopen* eröffnet worden sein.
- Die Datei darf nicht im Binärmodus eröffnet worden sein.
- Die Umgebungsvariable *IO\_CONVERSION* existiert nicht oder hat den Wert *YES*.

Außerdem bietet CRTE ab Version 2.0A eine *explizite* Konvertierung durch die Bibliotheksfunktionen *ascii\_to\_ebcdic* und *ebcdic\_to\_ascii* an.

- Beim Zugriff aus SINIX sind keine Maßnahmen erforderlich.

## Zugriff auf POSIX-Dateien

Auf POSIX-Dateien können Sie über POSIX-Programmschnittstellen zugreifen (siehe Abschnitt „POSIX-Programmschnittstellen“ auf Seite 53). Mehrere BS2000-Softwareprodukte unterstützen einen Zugriff auf POSIX-Dateien (siehe Kapitel „BS2000-Softwareprodukte im Umfeld von POSIX“ auf Seite 59).

## Zugriff auf ferne Dateien

Mit POSIX können Sie nur auf POSIX-Dateisysteme zugreifen, die sich am lokalen Rechner befinden. Um auch mit den Dateisystemen eines fernen Rechners arbeiten zu können, muß das Softwareprodukt NFS (Network File System) auf dem fernen und lokalen Rechner installiert sein. Am fernen Rechner (NFS-Server) muß das einzuhängende Dateisystem mit dem NFS-Kommando *share* bereitgestellt und am lokalen Rechner (NFS-Client) mit dem NFS-Kommando *mount* eingehängt werden. Danach kann auf das ferne Dateisystem vom lokalen Rechner aus zugegriffen werden. NFS gibt es für BS2000/OSD, UNIX, MS-DOS und Windows 95. NFS ist im Handbuch „NFS“ [12] beschrieben.

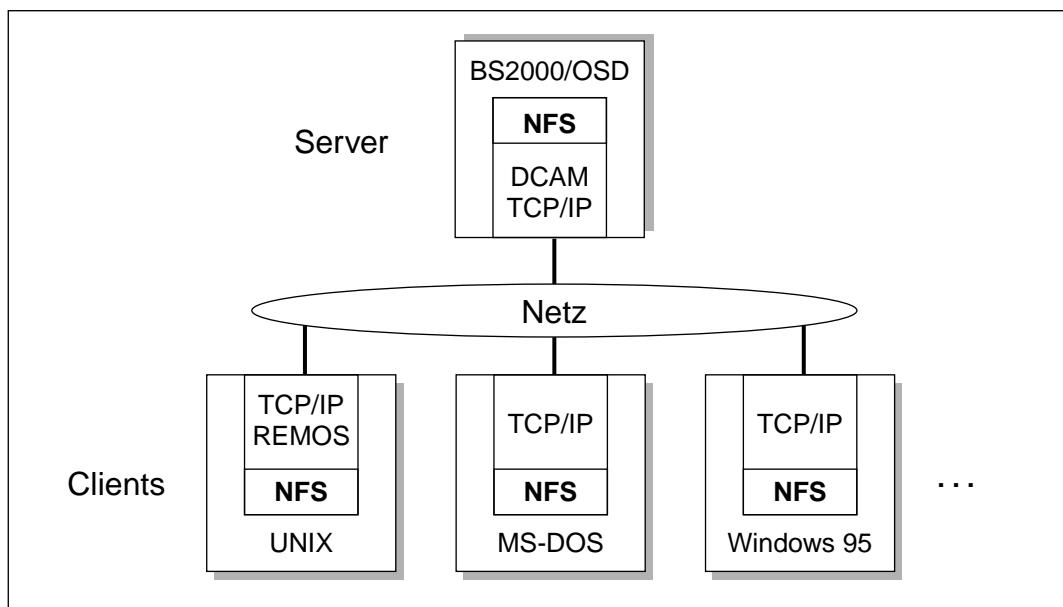


Bild 11: Verteilte Datenhaltung in einem heterogenen Rechnernetz mit NFS

### *r-Kommandos*

Das Kommando *rcp* (remote copy) ermöglicht das Kopieren von Dateien oder ganzen Dateibäumen zwischen POSIX-BS2000 und SINIX/UNIX. Auch das Kopieren zwischen zwei POSIX-BS2000-Systemen ist möglich, wenn eine TCP/IP-Verbindung besteht.

Bei *rcp* wird automatisch eine ASCII/EBCDIC-Konvertierung durchgeführt. Soll keine Konvertierung stattfinden (z.B. bei http-Dateien, die in POSIX im ASCII-Format benötigt werden), so muß *rcp* mit dem Schalter *-b* (binary) aufgerufen werden.

Mit dem Kommando *rsh* (remote Shell) können Kommandos auf einem SINIX/UNIX-Rechner ausgeführt werden.

Eine ausführliche Beschreibung der Kommandos *rcp* und *rsh* finden Sie im Handbuch „POSIX-Kommandos“ [2].

## **Migration von Mini-POSIX-Dateisystemen**

Mini-POSIX-Dateisysteme sind NFS- oder DFS-Dateisysteme, die mit BS2000/OSD V1.0 oder früheren BS2000-Versionen erstellt wurden. Sie können in POSIX-Dateisysteme migriert werden.

Mini-POSIX-Dateisysteme können entweder im Key-Format oder im Non-Key-Format angelegt worden sein. Wenn sie im Key-Format angelegt wurden, muß sie der Systemverwalter mit dem Programm *PAMCONV* in das Non-Key-Format umwandeln. Anschließend können sie als POSIX-Dateisysteme eingehängt werden. Die unter Mini-POSIX angelegten Dateien können als POSIX-Dateien bearbeitet werden.

## 2.3 POSIX als Subsystem im BS2000

POSIX ist ein privilegierter BS2000-Subsystem, das die Aufträge privilegierter und nicht-privilegierter Benutzer bearbeitet. Das Subsystem POSIX besteht im wesentlichen aus drei Teilen:

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

Das Subsystem POSIX unterstützt das POSIX-Dateisystem.

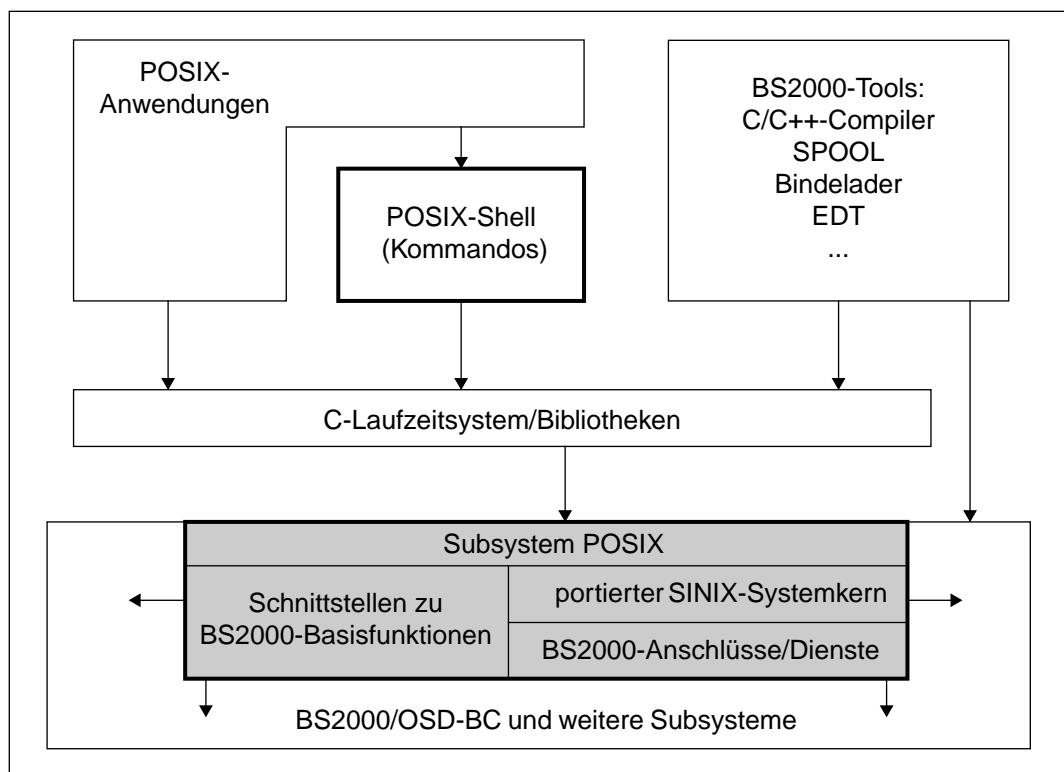


Bild 12: Einbettung des Subsystems POSIX in das BS2000

Allgemeine Informationen zu Subsystemen im BS2000 finden Sie im Handbuch „Verwaltung von Subsystemen“ [28].

## Verwaltung des Subsystems POSIX durch DSSM

Die Dynamische Subsystem-Verwaltung (DSSM) des BS2000/OSD bindet das Subsystem POSIX aus Bindelademodulen, die in der Programmbibliothek SYSLNK.POSIX-BC.<version> bereitgestellt werden.

Der Systemkern-Code von POSIX enthält - ebenso wie der Systemkern-Code des Original-UNIX - einige Steuerparameter (Tuningparameter), die ein Systemverwalter in der POSIX-Informationsdatei SYSSI.POSIX-BC.<version> entsprechend dem speziellen Einsatzfall setzen kann. Diese Steuerparameter dienen zum Konfigurieren des Systemkerns und zur Verbesserung der Performance. Außerdem wird in der POSIX-Informationsdatei der Name des Root-Dateisystems festgelegt.

In POSIX wird der UNIX-Tuningmechanismus auf den Parameterservice von DSSM abgebildet. Die POSIX-Informationsdatei enthält neben den Steuerparametern des Systemkern-Codes den Namen des Root-Dateisystems und andere POSIX-spezifische Steuerparameter. Der Inhalt der POSIX-Informationsdatei ist auf Seite 101 beschrieben.

Die POSIX-Informationsdatei wird zusammen mit anderen Komponenten an den Kunden ausgeliefert. Sie ist als SAM-Datei eingerichtet und enthält bereits Standardwerte. Die Standardwerte sind so gewählt, daß das Subsystem POSIX in beliebiger Umgebung ablaufen kann, ohne das Gesamtsystem durch übermäßigen Ressourcenverbrauch zu belasten. In vielen Fällen wird es aber sinnvoll sein, einige Steuerparameter, wie z.B. die maximale Anzahl von POSIX-Prozessen, an die spezielle POSIX-Anwendung und den Ressourcenvorrat des Gesamtsystems anzupassen.

Wenn der Systemverwalter einen ungültigen Parameterwert in die POSIX-Informationsdatei einträgt, wird die Meldung POS1020 ausgegeben. Statt des ungültigen Parameterwertes wird der Standardwert in die subsysteminterne Parametertabelle eingetragen.

Wenn beim Start des Subsystems POSIX keine POSIX-Informationsdatei vorhanden ist oder wenn sie nicht eröffnet werden kann, wird eine entsprechende Meldung ausgegeben und das Subsystem nicht gestartet.

## POSIX-Prozeßverwaltung

In POSIX findet der Programmablauf in einem Prozeß statt, im BS2000 in einer Task. POSIX-Prozesse werden auf BS2000-Tasks abgebildet.

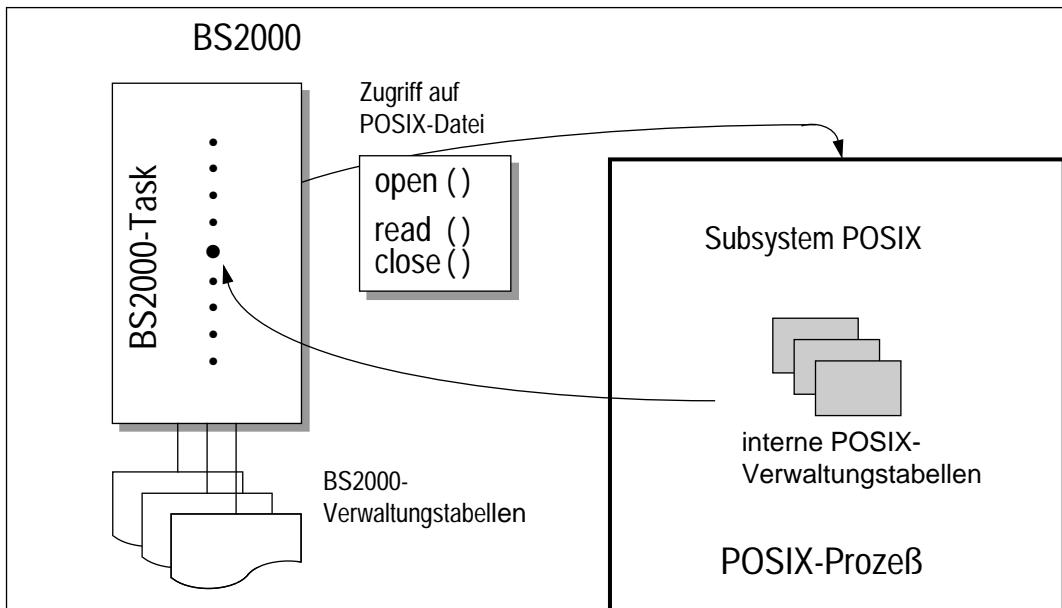


Bild 13: Zugriff einer BS2000-Task auf POSIX

In POSIX sind sämtliche Prozesse hierarchisch strukturiert. Die Prozeß-Hierarchie entsteht durch einen initiierenden Prozeß (*init*) und weitere Prozesse, die diesem initiierenden Prozeß untergeordnet sind. Man spricht von einer Vater-Sohn-Beziehung. Der initiierende Prozeß ist der Vater aller Prozesse. Die direkt untergeordneten Prozesse sind die Söhne, die wiederum Söhne haben können. Diese Rangfolge kann bis zu einer konfigurierbaren maximalen Prozeßanzahl (Steuerparameter NPROC in der POSIX-Informationsdatei) fortgeführt werden.

Im Folgenden sind die einzelnen POSIX-Prozesse näher beschrieben.

## fork

Durch den Aufruf der Funktion *fork* wird von einem Vaterprozeß ein neuer Sohnprozeß erzeugt. Die Funktion *fork* erzeugt eine neue Prozeßumgebung und kopiert ausgewählte Informationen des Vaterprozesses für den Sohnprozeß. Dem Sohnprozeß steht ein eigener, vom Vaterprozeß abgesonderter Adreßraum zur Verfügung. Der Sohnprozeß kann auf alle vom Vaterprozeß geöffneten POSIX-Ressourcen zugreifen.

Die beiden Prozesse laufen direkt nach dem Funktionsaufruf unabhängig voneinander weiter. Sie können durch den jeweiligen Returncode unterschieden werden: Der Sohnprozeß erhält den Wert 0 zurück, der Vater die Prozeßidentifikation (PID) des Sohnprozesses.

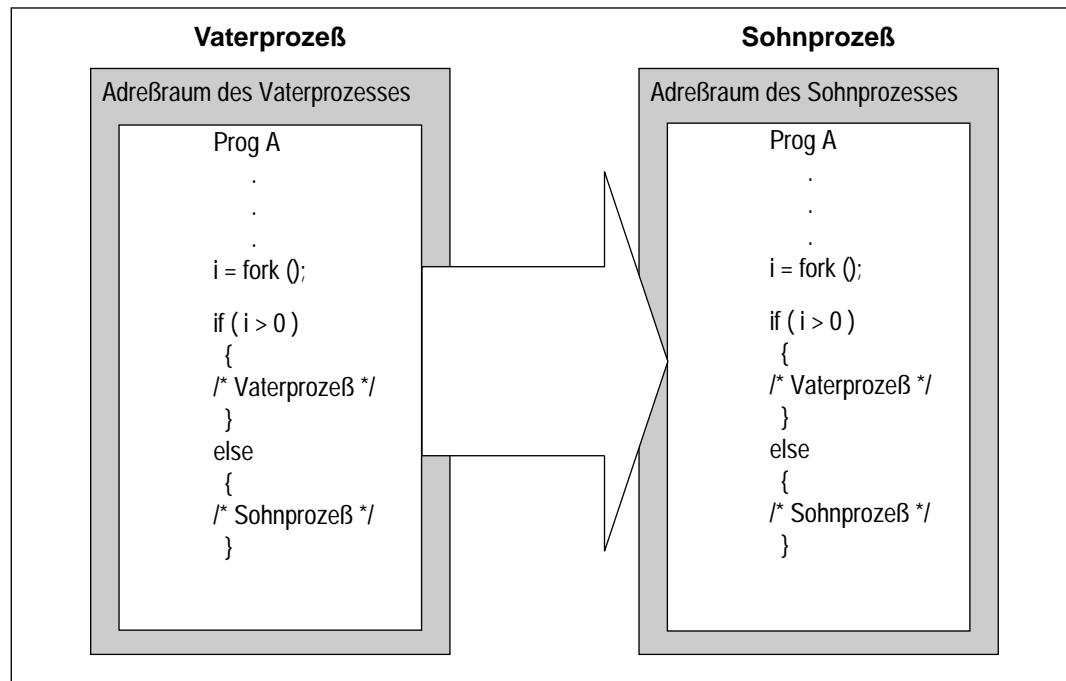


Bild 14: Arbeitsweise der Funktion *fork*



Bei einem Aufruf der Funktion *fork* werden DVS-Dateien und weitere BS2000-Betriebsmittel nicht mitvererbt.

### exec

Wenn in einem Programm die Funktion *exec* aufgerufen wird, wird die aktuelle Prozeßumgebung von einer neuen vollkommen überlagert. Dadurch kann z.B. in einem Sohnprozeß ein anderes Programm ablaufen als im Vaterprozeß. Die Prozesse bleiben aber durch die Vater-Sohn-Beziehung weiterhin verbunden.

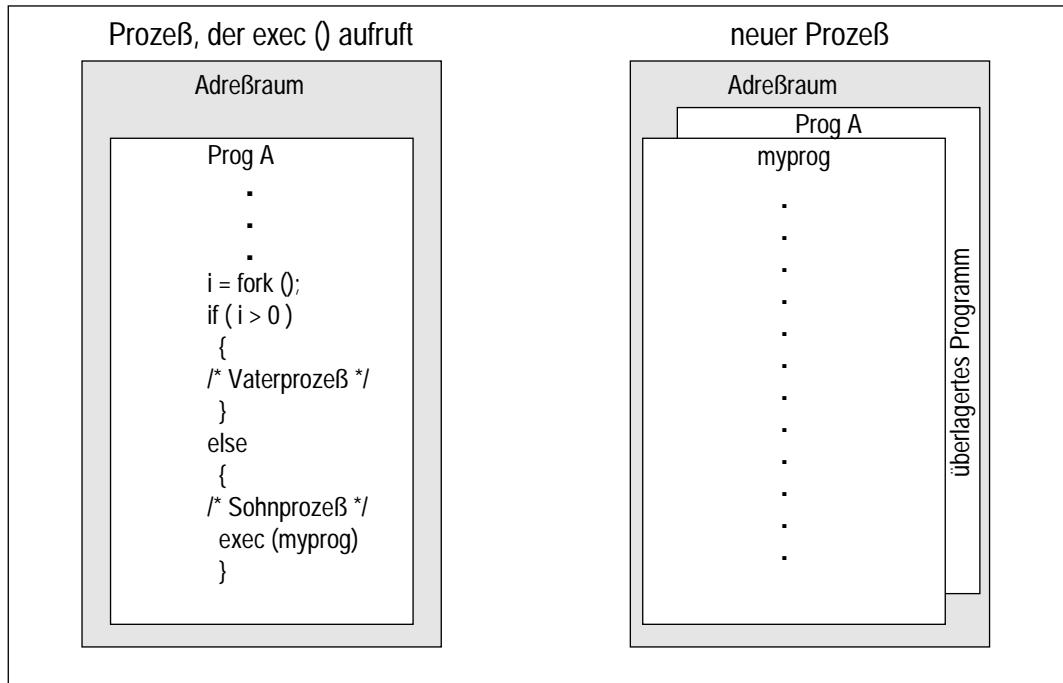


Bild 15: Arbeitsweise der Funktion exec

### Kombination von fork und exec

Die Funktionen *fork* und *exec* können auch kombiniert werden. Der Vorteil einer solchen Kombination besteht darin, daß Teilaufgaben auf einen anderen Prozeß ausgelagert werden können. Nachdem alle Teilaufgaben beendet sind, kann der Prozeß beendet werden.

Ein Beispiel dafür ist die POSIX-Shell. Die POSIX-Shell startet für einige Kommandos einen neuen Prozeß, der sich selbst überlagert und das kommandoausführende Programm startet. Nachdem dieses Programm beendet ist, wird der dafür erzeugte Prozeß beendet und die POSIX-Shell fortgesetzt.

## pipe

Anwendungsprogrammierern steht mit POSIX die Funktion *pipe* zur Interprozeß-Kommunikation zur Verfügung. Mit der Funktion *pipe* wird eine Pipe erzeugt. Eine Pipe ist ein Datenbehälter vom Typ „first in - first out“. Ein Prozeß kann eine Pipe verwenden, um Informationen an einen anderen Prozeß zu senden.

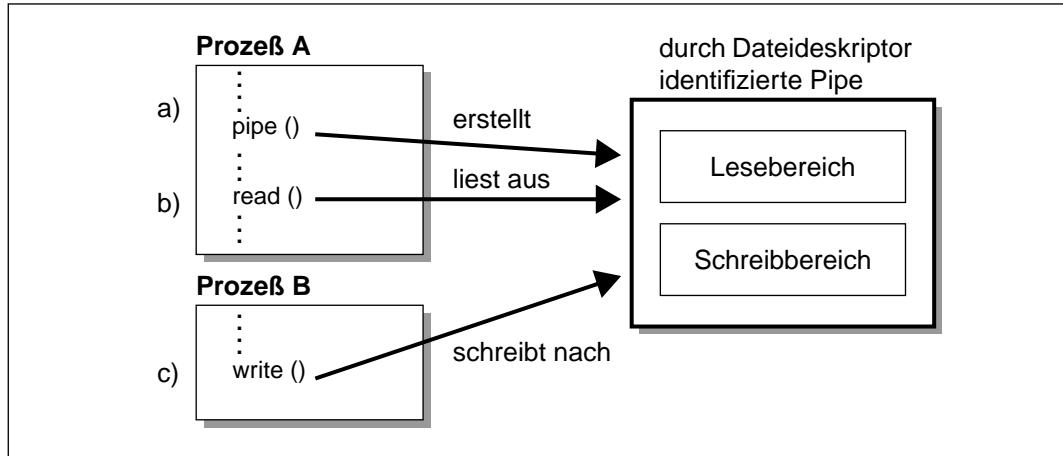


Bild 16: Arbeitsweise der unbenannten Pipe

- Durch den Aufruf der Funktion *pipe* durch Prozeß A wird eine Pipe zum Lesen und Schreiben von Daten erstellt. Die Pipe wird durch einen Dateideskriptor identifiziert. Der Dateideskriptor kennzeichnet eine Datei als offen.
- Prozeß B kann später in die Pipe schreiben und eine Nachricht hinterlassen, die Prozeß A danach zu einem beliebigen Zeitpunkt lesen kann.
- Prozeß A ruft die Funktion *read* auf, spezifiziert den auf die Pipe bezogenen Dateideskriptor und liest die vom Prozeß B hinterlegte Nachricht.

### „copy-on-access“-Mechanismus

Beim Aufruf der Funktion *fork* wird die komplette Prozeßumgebung vom Vater auf den Sohn vererbt. Dabei kann das Kopieren des kompletten Adreßraums relativ lange dauern. Der POSIX-spezifische *copy-on-access*-Mechanismus, der ähnlich wie der „copy-on-write“-Mechanismus des Original-UNIX funktioniert, verbessert die Performance erheblich: Bei einem *fork*-Aufruf wird der Adreßraum nicht kopiert, sondern lediglich markiert. Somit steht der Speicher dem Vater- und dem Sohnprozeß gemeinsam zur Verfügung. Wenn einer der beiden Prozesse zum ersten Mal auf eine Seite zugreift, wird anhand der Markierung festgestellt, daß eine Aktion auszuführen ist. In diesem Fall wird die betreffende Speicherseite für den Sohnprozeß kopiert, so daß beide Prozesse ihre eigene Version besitzen. Dadurch wird erreicht, daß nur die Speicherseiten kopiert werden, die tatsächlich benötigt werden. Der *fork*-Aufruf kann somit wesentlich performanter ablaufen.

Ein weiterer Vorteil des *copy-on-access*-Mechanismus besteht darin, daß häufig nach einem *fork*-Aufruf auch ein *exec*-Aufruf folgt. In diesem Fall ist das Kopieren des Speicherbereichs überflüssig, da er beim *exec*-Aufruf überlagert wird. Ohne *copy-on-access* würde der gesamte Speicherbereich kopiert und sofort wieder überschrieben.

### Holdertask

POSIX benutzt eine Holdertask. Die Initialisierung und Terminierung laufen unter Kontrolle dieser Holdertask ab. Während der Subsystem-Sitzung, d.h. zwischen Initialisierung und Terminierung, steht die Holdertask dem Subsystem POSIX nicht zur Verfügung.

### Aufrufertask

Die Aufrufertask wird beim ersten POSIX-Systemaufruf mit dem Subsystem POSIX verbunden, bei Programmende wird sie getrennt. Beim ersten POSIX-Systemaufruf erhält die Aufrufertask eine POSIX-Prozeßumgebung.

### Subsystem-private Server

Im Subsystem POSIX gibt es zwei Arten von subsystem-privaten Servern: Systemprozesse und Dämonen. Systemprozesse sind als BS2000-Systemtasks realisiert, Dämonen als nichtprivilegierte BS2000-Tasks.

Während das Kommando /START-SUBSYSTEM SUBSYSTEM-NAME=POSIX verarbeitet wird, werden nur Systemprozesse als Server initialisiert. Erst anschließend werden Dämonen erzeugt.

## 2.4 Sicherheitskonzept

In diesem Abschnitt ist beschrieben, wie POSIX in das BS2000 eingebettet wurde, um die Sicherheit des Gesamtsystems zu gewährleisten. Die nötigen Funktionen wurden zum Teil mit dem SINIX-Systemkern portiert, zum Teil sind sie Bestandteil des BS2000-Bausteins SRPM (System Resources and Privileges Management). Näheres zu SRPM finden Sie im Kapitel „POSIX-Benutzer verwalten“ auf Seite 113.

Das Sicherheitskonzept umfaßt:

- die Benutzerdatenverwaltung
- die Gruppenverwaltung
- den Zugriffsschutz für Behälterdateien
- den Zugriffsschutz für Dateien und Dateiverzeichnisse
- den Zugangsschutz bei Zugang über einen fernen Rechner

### Benutzerdatenverwaltung

Im POSIX-Standard sind Schnittstellen für die Sicherheitskontrolle eines Benutzers definiert. Mit diesen Schnittstellen werden feste Informationen über einen Benutzer erfragt, bevor er ein Betriebssystem benutzen darf. Folgende Benutzerdaten stehen für die Authentisierung zur Verfügung:

- Benutzerkennung/Login-Name des Benutzers
- Kennwort
- Benutzernummer
- Gruppennummer
- Initialwert für das Arbeitsverzeichnis
- zu startendes Programm

Diesen Informationen können nach Bedarf noch weitere hinzugefügt werden.

Beim Anmelden eines Benutzers im System wird die eingegebene Benutzerkennung und das zugehörige Kennwort gegen diese Informationen geprüft. Wenn die eingegebenen Werte stimmen, erhält der Benutzer Zutritt zum Betriebssystem. Näheres dazu finden Sie im Abschnitt „POSIX-Benutzerattribute vergeben“ auf Seite 117.

Zwischen den POSIX-Benutzerdaten und den BS2000-Benutzerdaten bestehen folgende Beziehungen:

- Der Login-Name des Benutzers und die BS2000-Benutzerkennung sind gleich.
- Das POSIX-Kennwort und das BS2000-LOGON-Kennwort sind gleich.

Zu den übrigen POSIX-Benutzerdaten gibt es auf der BS2000-Seite kein Äquivalent.

Für den Login-Namen gibt es in POSIX die Einschränkung, daß nur Großbuchstaben in Benutzernamen unterstützt werden.

Die POSIX-Benutzerdaten werden von der BS2000-Benutzerverwaltung gespeichert und verwaltet (näheres dazu siehe Kapitel „POSIX-Benutzer verwalten“ auf Seite 113). Sie sind als POSIX-Benutzerattribute in die BS2000-Benutzerdaten integriert. Der Zugriff auf POSIX-Benutzerdaten wird über die Benutzer- und Systemverwalterkommandos des BS2000 abgewickelt.

## Gruppenverwaltung

Die Gruppenverwaltung in POSIX entspricht der im UNIX. Sie unterscheidet sich gegenüber der Gruppenverwaltung im BS2000 in folgenden Punkten:

- In POSIX dienen Gruppen ausschließlich der Zuteilung von Zugriffsrechten auf Dateien. Im BS2000 haben Gruppen zusätzlich den Zweck, den Verbrauch von Ressourcen wie Plattspeicher, Rechnerleistung usw. zu steuern.
- In POSIX kann ein Benutzer gleichzeitig maximal 16 Gruppen angehören, im BS2000 nur einer einzigen Gruppe.
- BS2000-Gruppen sind hierarchisch angeordnet; in POSIX gibt es dieses Merkmal nicht.
- In POSIX kann ein Benutzer die aktuelle Gruppe wechseln; im BS2000 ist dies nicht möglich.

Wegen dieser großen Unterschiede existieren POSIX- und BS2000-Gruppen nebeneinander. Die POSIX- und BS2000-Gruppen werden getrennt verwaltet: Die POSIX-Gruppen auf der Shell-Ebene, die BS2000-Gruppen auf der BS2000-Ebene. Dies entspricht den unterschiedlichen Schutzmechanismen von POSIX- und BS2000-Dateien.

POSIX- und BS2000-Gruppen können unter Verzicht auf die Hierarchie identisch definiert werden, d.h. sie enthalten dann dieselben Benutzer.

Nähere Informationen zur Gruppenverwaltung finden Sie im Abschnitt „BS2000- und POSIX-Gruppen verwalten“ auf Seite 120.

## Zugriffsschutz für Behälterdateien

POSIX-Dateisysteme werden in Behälterdateien (Container) abgelegt. Behälterdateien sind BS2000-PAM-Dateien im Non-Key-Format. Sie sind vor unberechtigtem Zugriff über die Standard-Zugriffskontrolle des BS2000 (ACCESS-/USER-ACCESS-Attribute) geschützt.

Aus Sicherheitsgründen wird die Behälterdatei eines Benutzer-Dateisystems unter einer Systemkennung angelegt. Das LOGON-Kennwort einer Systemkennung ist nur dem POSIX-Verwalter bekannt.

Das POSIX-Installationsprogramm legt Behälterdateien als nicht mehrfach benutzbar und mit ACCESS=\*WRITE an. Diese Schutzattribute dürfen nicht verändert werden. Außerdem darf kein Dateikennwort vergeben werden.

Der Benutzer einer POSIX-Datei benötigt kein Zugriffsrecht für die Behälterdatei.

## Zugriffsschutz für Dateien und Dateiverzeichnisse

Der Zugriffsschutz für Dateien und Dateiverzeichnisse ist in POSIX durch folgende Schutzmechanismen realisiert:

- Benutzerkennungen
- Kennworte für Benutzerkennungen
- Zusammenfassen von Benutzerkennungen zu Gruppen
- Schutzbits für Dateien und Dateiverzeichnisse

Diese Schutzmechanismen verhindern, daß ein Benutzer die Dateien und Dateiverzeichnisse eines anderen Benutzers unberechtigt lesen und verändern kann.

### Zugriffsschutz durch Benutzerkennung, Kennwort und Gruppennummer

Jeder, der POSIX benutzen will, benötigt am entsprechenden BS2000-Rechner eine Benutzerkennung, die der BS2000-Systemverwalter einrichten muß. Der Benutzer selbst kann ein Kennwort festlegen oder verändern, um seine Benutzerkennung vor unberechtigtem Zugriff zu schützen.

Siehe auch Abschnitt „Zugangsschutz bei Zugang über einen fernen Rechner“ auf Seite 44.

Benutzer können zu Gruppen zusammengefaßt werden. Dadurch können Dateien und Dateiverzeichnisse allen Mitgliedern dieser Gruppe zugänglich gemacht werden. Der Systemverwalter muß dazu jedem Benutzer eine Gruppennummer zuordnen. Benutzer mit der gleichen Gruppennummer gehören der gleichen Gruppe an (siehe Abschnitt „BS2000- und POSIX-Gruppen verwalten“ auf Seite 120).

## Zugriffsschutz durch Schutzbits

Jeder Datei und jedem Dateiverzeichnis werden beim Erstellen automatisch die Benutzer- und Gruppennummer des erstellenden Prozesses und Schutzbits zugewiesen. Diese Schutzbits sind für bestimmte Zugriffe standardmäßig vorbelegt.

Es gibt Schutzbits für die folgenden drei Benutzerklassen:

- Eigentümer der Datei
- Gruppe, der der Eigentümer angehört
- Andere

Jede dieser Benutzerklassen besitzt je ein Schutzbitt für Leseberechtigung (read), Schreibberechtigung (write) und Ausführberechtigung (execute).

### Beispiel

Eigentümer: r w x

Gruppe: r w -

Andere: r - -

Die Schutzbits gelten ausschließlich für ihre Benutzerklasse. Wenn z.B. nur der Eigentümer eine Zugriffsberechtigung für eine Datei besitzt, darf weder die Benutzerklasse *Gruppe* noch die Benutzerklasse *Andere* mit dieser Datei arbeiten.

Die Zugriffsberechtigungen haben für Dateien und Dateiverzeichnisse unterschiedliche Bedeutung:

Zugriffsberechtigung	Datei	Dateiverzeichnis
read	lesen	Einträge lesen
write	schreiben	Einträge (Dateien) löschen/anlegen
execute	ausführen	durchlaufen/durchsuchen

Vor dem ersten Schutzbitt für den Eigentümer steht noch ein Identifikationszeichen, das automatisch vergeben wird. Es hat folgende Bedeutung:

- Datei
- b blockorientiertes Gerät
- c zeichenorientiertes Gerät
- d Dateiverzeichnis
- I Symbolischer Verweis

Die Schutzbits können mit dem POSIX-Kommando *chmod* geändert werden. Ein Benutzer mit der Benutzernummer 0 kann die Schutzbits von allen Dateien und Dateiverzeichnissen ändern, der Eigentümer nur von seinen eigenen Dateien und Dateiverzeichnissen. Auch wenn jemand aus der Benutzerklasse *Gruppe* oder *Andere* volle Zugriffsberechtigung auf eine Datei oder ein Dateiverzeichnis besitzt, kann er die Schutzbits nicht ändern.

Die Schutzbits für die Benutzerklasse *Gruppe* werden entsprechend der Gruppenzugehörigkeit des Eigentümers vergeben. Beim Anlegen einer neuen Datei wird die Gruppennummer und damit die Gruppenzugehörigkeit vom aktuellen Dateiverzeichnis übernommen.

Die aktuell gültige Schutzbitt-Maske kann mit dem POSIX-Kommando *umask* ausgegeben oder geändert werden. Diese Schutzbitt-Maske legt fest, welche Zugriffsrechte die Dateien und Dateiverzeichnisse erhalten, die Sie ab jetzt in der aktuellen Shell oder in einer ihrer Subshells neu anlegen.

Wenn Sie mit *umask* die Schutzbitt-Maske ändern, gilt diese Änderung so lange, bis Sie entweder mit *umask* einen neuen Wert vereinbaren oder die Shell beenden, in der Sie *umask* aufgerufen haben.

POSIX-Verwalter können mit *umask* den Wert der Schutzbitt-Maske in der Datei */etc/profile* festlegen. Da die Datei */etc/profile* von jeder Login-Shell ausgeführt wird, gelten die so bestimmten Zugriffsrechte für jeden am System angemeldeten Benutzer.

Nähere Informationen zu den POSIX-Kommandos *chmod* und *umask* finden Sie im Handbuch „POSIX-Kommandos“ [2].

## Zugangsschutz bei Zugang über einen fernen Rechner

POSIX kann auch von fernen Rechnern aus genutzt werden (siehe Abschnitt „Zugang zur POSIX-Shell“ auf Seite 47). Benutzer, die sich mit dem Kommando *rlogin* an POSIX anschließen, sind wie lokale Benutzer in der BS2000-Benutzerverwaltung des Zentralrechners eingetragen. Der BS2000-Baustein SRPM überprüft die Zugangsberechtigung während der *rlogin*-Verarbeitung.

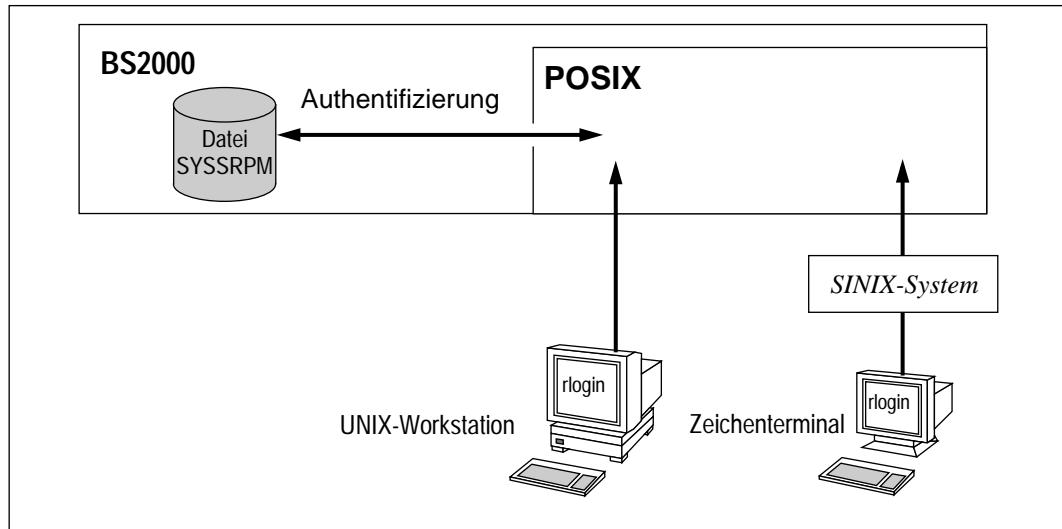


Bild 17: Zugangsschutz bei Zugang über *rlogin*

Für die Kommandos *rcp* und *rsh* gilt:

Die Zugangsberechtigungen werden wie in SINIX/UNIX geregelt, d.h. zugelassene Rechner und Benutzer werden der Datei *\$HOME/.rhosts* entnommen.

# 3 Arbeiten mit POSIX

Dieses Kapitel wendet sich an alle POSIX-Benutzer. Es informiert Sie über die POSIX-Shell und die POSIX-Programmschnittstellen. Außerdem enthält es eine Beispielsitzung.

## 3.1 POSIX-Shell

Die POSIX-Shell ist die Schnittstelle, die Sie - über das C-Laufzeitsystem/Bibliotheken - mit dem Subsystem POSIX verbindet. Das folgende Bild zeigt die Struktur von POSIX im BS2000 und die Einbettung der POSIX-Shell.

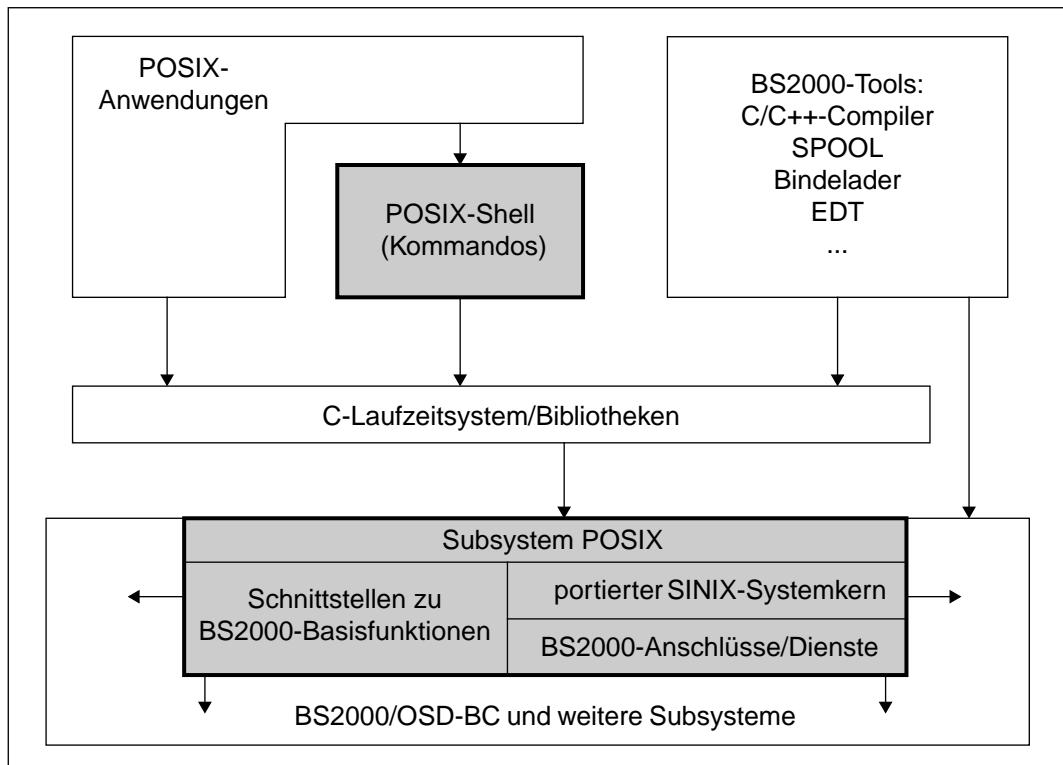


Bild 18: Struktur von POSIX im BS2000 und Einbettung der POSIX-Shell

Die POSIX-Shell ist eine Kommandoschnittstelle, die Sie zusätzlich zur Kommandoschnittstelle des BS2000/OSD verwenden können (siehe Bild 19).

Nach erfolgreichem Zugang zur POSIX-Shell (siehe Seite 47) stehen Ihnen alle Kommandos der POSIX-Shell zur Verfügung. Nach dem Verlassen der POSIX-Shell können Sie wieder BS2000-Kommandos eingeben.

In allen POSIX-Versionen steht eine Basis-Shell zur Verfügung, die Bestandteil von POSIX-BC ist (siehe Abschnitt „Lieferumfang“ auf Seite 77). Der Funktionsumfang dieser Basis-Shell ist auf Seite 179 beschrieben.

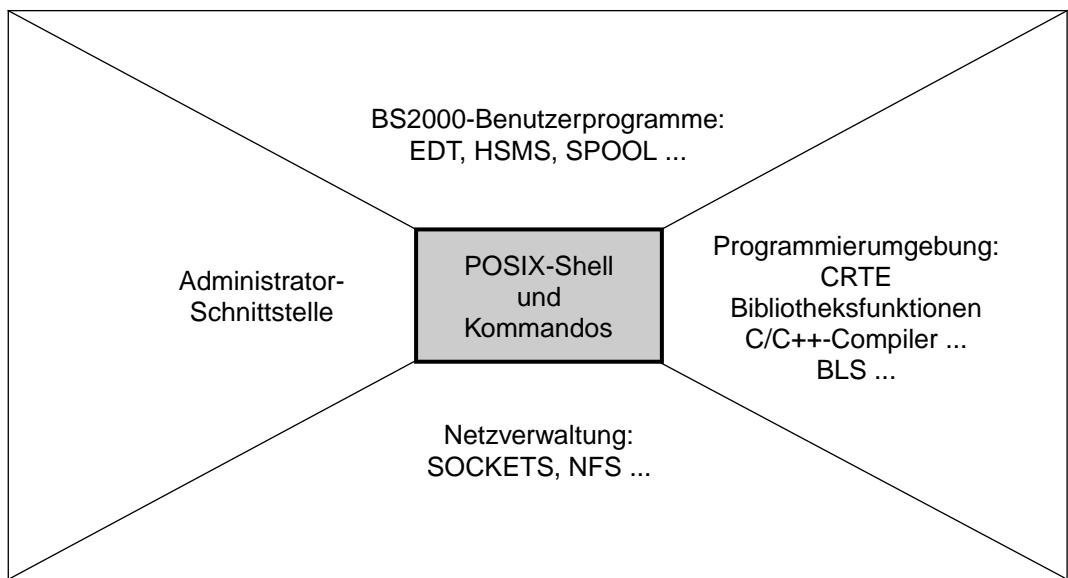


Bild 19: Die Komandoebene POSIX-Shell im Subsystem POSIX

Die POSIX-Shell liest Kommandos von einer Datensichtstation oder aus einer UFS-Datei, interpretiert sie nach bestimmten Regeln und sorgt für die Ausführung. Eine Datei, die Kommandos für die POSIX-Shell enthält, heißt Shell-Prozedur (Shell-Skript).

Die Bedienung und Leistung der POSIX-Shell hängen davon ab, ob die Datensichtstation, an der der Benutzer arbeitet, ein Block- oder Zeichenterminal ist.

Die POSIX-Shell bietet Ihnen eine umfangreiche Kommandosprache, die sich wie eine Programmiersprache anwenden lässt. Sie können mit den vorhandenen Kommandos eigene Programme erstellen und ohne vorheriges Übersetzen ausführen.

### 3.1.1 Zugang zur POSIX-Shell

Zur POSIX-Shell gibt es folgende Zugangsmöglichkeiten:

- über ein BS2000-Terminal (Blockterminal)
- von einem UNIX- oder SINIX-Terminal (Zeichenterminal)
- über eine Terminal-Emulation

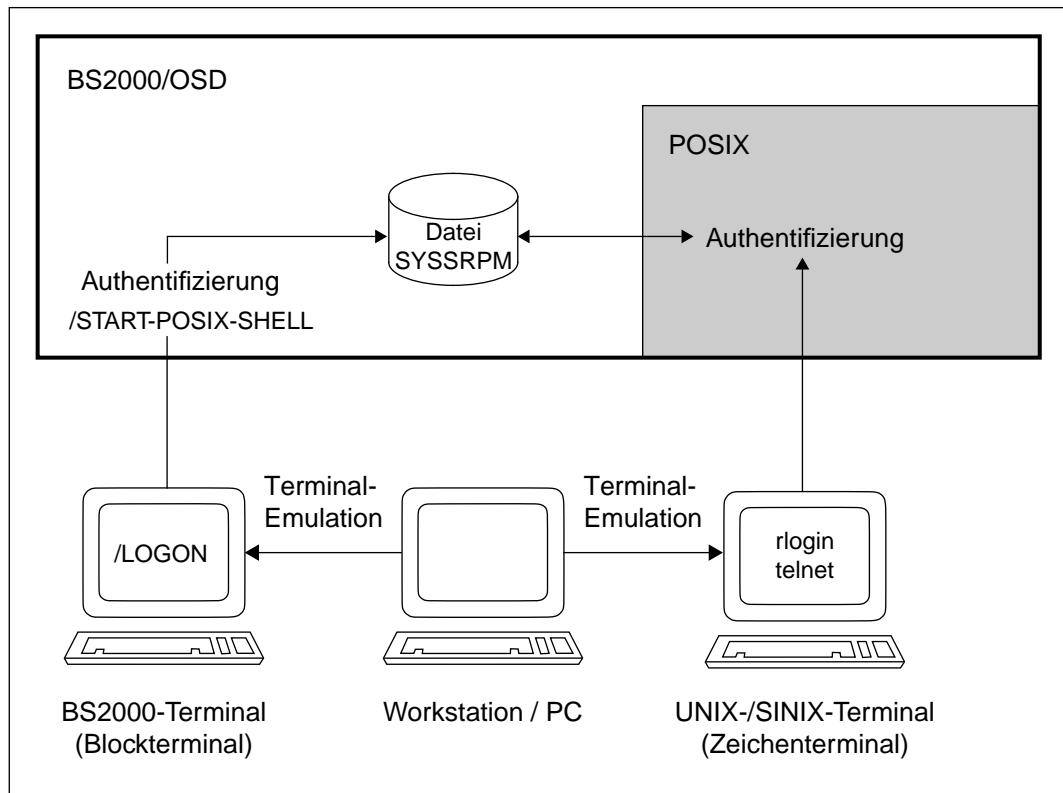


Bild 20: Zugangsmöglichkeiten zur POSIX-Shell

#### Zugang von einem Blockterminal

Jeder BS2000-Benutzer kann nach dem erfolgreichen BS2000-LOGON mit dem BS2000-Kommando **/START-POSIX-SHELL** (siehe Seite 157) die POSIX-Shell starten. Dieses Kommando besitzt keine POSIX-relevanten Operanden, da lediglich die POSIX-Umgebung aufgebaut und das Programm aufgerufen wird, das in der Datei **SYSSRPM** für den betreffenden Benutzer eingetragen ist (siehe Benutzerattribut „Programm“ im Abschnitt „POSIX-Benutzerattribute vergeben“ auf Seite 117).

Wenn ein Benutzer die POSIX-Shell als Standardprogramm in seinen Benutzerdaten eingetragen hat, kann er interaktiv mit der POSIX-Shell arbeiten, nachdem er das BS2000-Kommando /START-POSIX-SHELL eingegeben hat. Ihm stehen dann alle Kommandos und Funktionen der POSIX-Shell zur Verfügung. Für Interaktionen zwischen dem BS2000 und dem Subsystem POSIX gibt es in der POSIX-Shell eigene Kommandos (siehe Abschnitt „Kommandoumfang der Basis-Shell“ auf Seite 179).

Eine Rückkehr ins BS2000 ist nur möglich, indem der Benutzer mit dem POSIX-Kommando *exit* die POSIX-Shell beendet.

## Zugang von einem Zeichenterminal

### Zugang über *rlogin*

Der Benutzer kann sich von einem UNIX- oder SINIX-Terminal mit dem Kommando *rlogin* an einem BS2000-Rechner anmelden, wenn der Benutzer für den BS2000-Rechner eine Zugangsberechtigung besitzt. Dazu benötigt er am BS2000-Rechner eine Benutzerkennung mit dem zugehörigen Kennwort. Nach dem Anmelden kann er POSIX wie im lokalen Modus benutzen.

Um sich an einem BS2000-Rechner anzuschließen, muß der Benutzer folgendes Kommando in der Shell eingeben:

```
rlogin <host> [-l <benutzerkennung>]
```

Wenn der Benutzer keine Benutzerkennung eingibt, wird die Kennung verwendet, unter der er am lokalen Rechner angemeldet ist. Beim *rlogin* wird für die gewünschte Benutzerkennung das Kennwort erfragt. Das Kennwort wird über die BS2000-Komponente SRPM (**S**ystem **R**esources and **P**rivileges **M**anagement) verifiziert: Die Angaben für die BS2000-Benutzerkennung und das Kennwort werden gegen die im Home-Pubset gesicherten Zugangskontrollattribute geprüft. Wenn Übereinstimmung besteht, erhält der Benutzer Zugang zum Subsystem POSIX. Wenn das Produkt SECOS im Einsatz ist, kann die Zugangskontrolle über die LOGON-Protection noch weiter verfeinert werden.

Für die Abrechnung eines Remote-Login-Systemlaufs muß eine Abrechnungsnummer vorhanden sein. Diese kann beim Kommando ADD-USER (siehe Seite 127) mit dem Operanden POSIX-RLOGIN-DEFAULT festgelegt werden.

Beim Zugang über *rlogin* kann der Benutzer nicht auf BS2000-Kommandos zugreifen.

### *Zugang über telnet*

Mit dem *telnet*-Dämon *telnetd* wird ein direkter Zugang zum BS2000 über das *telnet*-Protokoll sowohl vom SINIX-/UNIX-System aus als auch vom PC direkt über die *telnet*-Anwendung realisiert, die sich gegenüber POSIX wie ein Zeichenterminal verhält. Die Zugangskontrolle erfolgt in derselben Weise wie bei *rlogin*, d.h. über BS2000-Zugangsmechanismen. Der Zugang ohne Angabe eines Kennwortes, wie es zwischen UNIX-Systemen möglich ist (durch einen Eintrag in der *.rhosts*-Datei), wird nicht unterstützt.

Der dazugehörige Dämon *in@telnetd* wird in POSIX nur dann automatisch gestartet, wenn in der POSIX-Datei */etc/inetd.conf* das Kommentarzeichen '#' vor 'telnet' entfernt wurde. Bitte beachten Sie dazu die Hinweise in der Freigabemitteilung zu POSIX-BC.

### **Zugang über eine Terminal-Emulation**

Die dritte Zugangsmöglichkeit besteht über eine Terminal-Emulation. Dazu meldet sich der Benutzer an einer Workstation oder an einem PC an. Anschließend startet er eine Terminal-Emulation, wobei entweder ein UNIX-Zeichenterminal oder ein BS2000-Blockterminal emuliert werden muß:

#### *BS2000-Terminal-Emulation*

Beim Zugang über eine BS2000-Terminal-Emulation wie z.B. EM9750 oder MT9750 steht dem Benutzer ein Terminal in der Art eines Blockterminals zur Verfügung. Der Benutzer muß sich wie im BS2000 üblich authentisieren und kann danach BS2000-Kommandos und /START-POSIX-SHELL wie über ein BS2000-Terminal eingeben (siehe Seite 47).

#### *UNIX-/SINIX-Terminal-Emulation*

UNIX-/SINIX-Terminal-Emulationen stehen für UNIX-/SINIX-Workstations mit grafischen, OSF/Motif-basierten Oberflächen und für Windows(95/NT)-PCs zur Verfügung (z.B. EM97801, SINIX-TE). Dabei wird ein UNIX-/SINIX-Zeichenterminal emuliert und der Benutzer kann UNIX-Kommandos wie z.B. *rlogin* eingeben (siehe Seite 48).

### 3.1.2 Besonderheiten für das Arbeiten mit der POSIX-Shell

#### Voreinstellungen in der Benutzerumgebung

Nach erfolgreichem Zugang zum Subsystem POSIX wird die POSIX-Shell gestartet. Bevor die POSIX-Shell ihr Bereitzeichen ausgibt, werden folgende Voreinstellungen in der Benutzerumgebung getroffen:

- Die POSIX-Shell initialisiert die Standard-Shell-Variablen. Sie weist den folgenden Variablen ihre Standardwerte zu:  
HOME, LANG, LOGNAME, MAIL, PATH, PS1, PS2, PS3, PS4, SHELL, TTY, TERM, TZ und USER.  
Falls eine Variable <x> durch die BS2000-S-Variablen SYSPOSIX.<x> bereits definiert ist, wird dieser Wert genommen. Es dürfen aber nicht die Shell-Variablen USER, TERM, TYP, LOGNAME, HZ, HOME und MAIL vom Anwender gesetzt werden.
- Die Datei */etc/profile* wird ausgeführt.
- Die Datei *\$HOME/.profile* wird ausgeführt, falls Sie diese Datei angelegt haben.

#### Sonderfunktionen (P-Tasten, Ctrl-Taste)

Sie können die P-Tasten **[P3]** und **[P4]** durch den Aufruf des POSIX-Kommandos *bs2pkey* folgendermaßen belegen:

**[P3]** mit @@c ([**CTRL**] **C**)  
**[P4]** mit @@d ([**CTRL**] **D**)

Das Programm wird entweder in der POSIX-Shell (ohne Optionen) aufgerufen oder kann in die Datei */etc/profile* aufgenommen werden. Dann wird das Programm bei jedem Aufruf der Shell aktiviert.

### 3.1.3 POSIX-Lader

Der POSIX-Lader ist Bestandteil des Subsystems POSIX. Er verwaltet einen Memory-Pool variabler Größe, in dem vorgeladene POSIX-Tools bereitgestellt werden. Mit Hilfe des POSIX-Laders können Ladevorgänge in POSIX wesentlich beschleunigt werden.

Der POSIX-Lader besteht aus zwei Teilen:

- LINKER:
- Lädt das Programm über BLS-Schnittstellen (Binder-Lader-System)
  - Speichert das vorgeladene Programm im Memory-Pool
  - Startet das Programm
- LOADER:
- Kopiert das Programm aus dem Memory-Pool in den Speicher
  - Startet das kopierte Programm

Beim Start von POSIX ist der Memory-Pool leer. Beim ersten Aufruf eines POSIX-Tools über den Systemaufruf *exec* wird der LINKER aktiv.

Dieser lädt das Programm „klassisch“ über BLS-Schnittstellen. Vor dem Start des Programmes sichert der LINKER das vorgeladene Programm in seinem Memory-Pool.

Bei jedem weiteren Aufruf eines POSIX-Tools über den Systemaufruf *exec* wird der LOADER aktiv und prüft folgende Bedingungen:

- Ist die Task eine durch *fork* erzeugte Task?
- Wird das Programm nicht debugged?
- Sind bereits vorgeladene Programme im Memory-Pool?

Sind alle Bedingungen erfüllt, wird das im Memory-Pool bereitgestellte vorgeladene Programm direkt (unter Umgehung von BLS) in den Speicher kopiert und gestartet.

Durch das Umgehen von BLS werden die vom LOADER gestarteten Programme unvollständig in die BS2000-Programmumgebung eingebettet. Das bringt Einschränkungen für Programme mit sich, die im Memory-Pool vorgeladen sind:

- Es wird keine Lademeldung ausgegeben.
- Der Programmname in der Ausgabe des Kommandos /SHOW-JOB-STATUS fehlt.
- Das Kommando *debug* verliert seine Funktion, wenn es im Memory-Pool geladen wird.

In der Informationsdatei (siehe Seite 101) müssen die Parameter DBLPOOL und DBLSTATE eingetragen sein.

### 3.1.4 Kommandos von der POSIX-Shell aus eingeben

Nach erfolgreichem Zugang zum Subsystem POSIX wird die POSIX-Shell gestartet.

Wenn Sie die POSIX-Shell interaktiv benutzen, gibt die POSIX-Shell den Wert der Umgebungsvariablen PS1 als Bereitzeichen aus, bevor sie ein Kommando einliest. Im Standardfall ist dies das Dollarzeichen (\$) bzw. (#) für einen privilegierten Benutzer und ein anschließendes Leerzeichen (...).

Die Kommandoeingabe hat folgendes Format:

`kommando[optionen][parameter] ... ↵`

Bei *kommando* müssen Sie den Namen eines POSIX-Kommandos oder einer Shell-Prozedur angeben, die ausgeführt werden soll. Mit *optionen* geben Sie Steueranweisungen zur Kommandoausführung. Bei *parameter* müssen Sie ein Aufrufargument eingeben, das die POSIX-Shell an *kommando* übergibt. Abhängig vom Kommando können Sie auch mehrere Aufrufargumente angeben.

Den Kommandonamen und die Aufrufargumente müssen Sie bei Zeichenterminals durch Tabulator- oder Leerzeichen voneinander trennen. Das letzte Aufrufargument und damit die Eingabe des Kommandos schließen Sie bei Zeichenterminals durch ↵ und bei Blockterminals durch **[EM] [DUE]** ab.

Das Starten von reinen BS2000-Programmen aus der POSIX-Shell wird nicht unterstützt.

Wenn die Bildschirmzeile für eine Eingabe zu kurz ist, haben Sie zwei Möglichkeiten:

- Sie schreiben am Zeilenende einfach weiter, ohne die Taste ↵ zu drücken. Nachdem Sie das Kommando vollständig eingegeben haben, schließen Sie es mit der Taste ↵ ab.
- Sie setzen die Zeile mit ☐ ↵ fort. Das Zeichen Gegenschrägstrich (\\) entwertet die Kommandoabschlußfunktion der Taste ↵. Anschließend können Sie die Kommando-eingabe fortsetzen. Nach dem Drücken von ↵ (ohne ☐) wird das Kommando ausgeführt.

Jedes POSIX-Kommando gibt an die POSIX-Shell, in der es aufgerufen wurde, einen Wert zurück, nämlich seinen Endestatus. Dieser Wert ist bei einem fehlerfreien Ablauf 0, bei einem fehlerhaften Ablauf ungleich 0.

Wenn ein Kommando Informationen auf den Bildschirm ausgibt und die Ausgabe größer als eine Bildschirmseite ist, können Sie bei Zeichenterminals die Ausgabe durch Drücken der Tasten **[CTRL] [S]** anhalten und anschließend mit **[CTRL] [Q]** fortsetzen. Bei Blockterminals wird diese Funktion nicht unterstützt.

Ausführliche Informationen zur Eingabe von Kommandos finden Sie im Handbuch „POSIX-Kommandos“ [2].

## 3.2 POSIX-Programmschnittstellen

Die POSIX-Programmschnittstellen stehen zusammen mit den BS2000-Programmschnittstellen zur Verfügung. Deshalb sind reine BS2000-Programme, reine POSIX-Programme und gemischte Programme ablauffähig. Gemischte Programme enthalten sowohl BS2000- als auch POSIX-Programmschnittstellen. Für gemischte Programme bestehen einige Einschränkungen (siehe Seite 54).

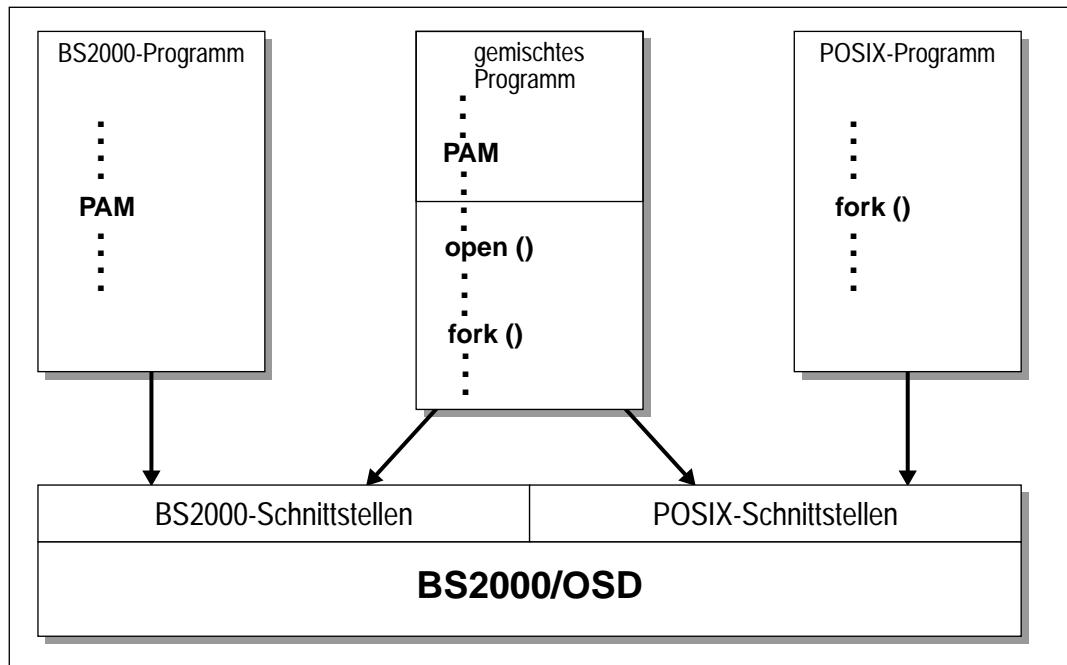


Bild 21: Reine und gemischte Anwendungen

Die POSIX-Programmschnittstellen enthalten C-Bibliotheksfunktionen mit BS2000- und POSIX-Funktionalität. Programme, die auf anderen Plattformen (UNIX/SINIX, DOS, ...) gemäß dem XPG4-Standard geschrieben wurden, müssen lediglich neu übersetzt werden, um im BS2000 ablaufen zu können.

Auf POSIX-Dateien kann über reine oder gemischte Programme zugegriffen werden. Darüber hinaus können POSIX-Dateien von einigen BS2000-Softwarereprodukten wie EDT und HSMS verarbeitet werden (siehe Kapitel „BS2000-Softwarereprodukte im Umfeld von POSIX“ auf Seite 59).

### 3.2.1 Einschränkungen für Programme mit gemischter Funktionalität

Programme, die POSIX-Schnittstellen nutzen, werden analog zu Programmen mit reinen BS2000-Schnittstellen behandelt. Einschränkungen ergeben sich dort, wo ein Prozeß mit einem *fork*-Aufruf erzeugt wird und ihm seine BS2000-Umgebung nicht mitvererbt wird.

Zwischen folgenden Aufrufen eines gemischten Programms muß unterschieden werden:

1. Logon-Prozeß (Dialog-Task), der nicht durch einen *fork*-Aufruf erzeugt wurde  
BS2000- und POSIX-Programmschnittstellen können beliebig gemischt werden.
2. Prozeß, der durch einen *fork*-Aufruf aus einem Logon-Prozeß erzeugt wurde

Die SYSFILE-Umgebung wird nicht weitervererbt, weil generell keine vom Vaterprozeß geöffneten BS2000-Dateien an den Sohnprozeß vererbt werden.

Die SYSFILE-Umgebung ist je eine anwenderspezifische Systemdatei zur Kommando-eingabe (SYSCMD), zur Dateneingabe (SYSDTA), zum Logging (SYSLST) und zur Meldungs- bzw. Datenausgabe (SYSOUT).

Eine Mischung von BS2000- und POSIX-Programmschnittstellen ist in folgendem Umfang erlaubt:

- Parallele Ausgabe über POSIX-Mechanismen und WROUT erlaubt
- Keine Eingabe über RDATA möglich
- Kein Checkpoint/Restart möglich
- Kein *BKPT* möglich
- Kein *fork* möglich, wenn DIV- oder FASTPAM-Bereiche existieren

Ansonsten können die BS2000- und POSIX-Programmschnittstellen beliebig gemischt werden.

3. Gemischtes Programm, das aus der POSIX-Shell gestartet wird

Ein gemischtes Programm, das aus der POSIX-Shell gestartet wird, hat eine andere SYSFILE-Umgebung als die POSIX-Shell, da es durch einen *fork*-Aufruf erzeugt wurde. Es gilt dasselbe wie bei 2.

### 3.2.2 Einschränkungen für Makroaufrufe

Jeder Prozeß, der durch einen Aufruf der Funktion *fork* erzeugt wurde, besitzt eine SYSFILE-Umgebung, aber keine Systemdatei SYSCMD (ausgenommen *rlogin*-Sessions). Deshalb werden Zugriffe auf die Systemdatei SYSCMD mit einem Returncode zurückgewiesen. Ansonsten können BS2000-Kommandos per CMD-Makro eingegeben werden, wobei folgendes zu beachten ist:

- Ein CMD-Makroaufruf erfordert eine nachträgliche SDF-Initialisierung der durch *fork* erzeugten Task. Dadurch verschlechtert sich die Performance.
- Folgende Kommandos sind in einem CMD-Makroaufruf nicht zugelassen:
  - Alle Kommandos, die bereits in BS2000/OSD-BC nicht zugelassen sind (siehe Handbuch „Makroaufrufe an den Ablaufteil“ [29]); u.a. ist das Verbot von /HOLD-PROGRAM zu beachten.
  - Kommandos, die auf die Systemdatei SYSCMD zugreifen, z.B. das %TRACE-Kommando von AID

Die BS2000-Kommandos /EXIT-JOB und /LOGOFF führen wie in BS2000/OSD-BC zur Beendigung des Tasks und zur Rückkehr zum Vaterprozeß.

### 3.2.3 Vererbung

Bei einem *fork*-Aufruf werden nur POSIX-Ressourcen vererbt. Deshalb sind POSIX-Dateien, die der Vaterprozeß eröffnet hat, auch in Sohnprozessen offen. Dagegen sind BS2000-Dateien, die im Vaterprozeß geöffnet wurden, nicht offen.

Der Klasse-6-Speicher eines Programms wird vollständig vererbt. Beim Klasse-5-Speicher werden nur die vorher als vererbbar markierten Seiten vererbt.

Eine Quasi-Vererbung von BS2000-Ressourcen ist programmgesteuert möglich, indem man im Vaterprozeß diese Ressourcen (z.B. BS2000-Dateien) mehrfach benutzbar öffnet. Die Information über diese Ressourcen kann dann über eine privat zu definierende Datenstruktur dem Sohnprozeß, in dem ja dasselbe Programm wie im Vaterprozeß abläuft, mitgegeben werden. Der Sohnprozeß kann sich dann an diese Ressourcen wieder explizit anschließen.

### 3.3 Beispielsitzung

In diesem Abschnitt finden Sie ein Beispiel für das Arbeiten mit der POSIX-Shell. Sie melden sich an das BS2000 an, lassen sich das Inhaltsverzeichnis Ihrer Benutzerkennung anzeigen und starten dann die POSIX-Shell.

In der POSIX-Shell erstellen Sie zuerst eine *.profile*-Datei, in der Sie zur Arbeitsvereinfachung Aliasvariablen und zur besseren Orientierung ein neues Bereitzeichen definieren, das den jeweils aktuellen Pfad ausgibt. Nach der Ausführung der *.profile*-Datei sind die dort getroffenen Definitionen wirksam.

Anschließend übertragen Sie eine Datei des BS2000-Dateisystems in das POSIX-Dateisystem und bearbeiten sie dort.

```
/set-logon-parameters user-id=user1,account=... _____ (1)
/show-file-attributes _____ (2)
%
% 114 :10SN:$USER1.ANHANG.V2
% 3 :10SN:$USER1.AVASQUER
% 78 :10SN:$USER1.BIB.EXAMPLES.SDF
% 6 :10SN:$USER1.DO.MSGCHECK
% 5007 :10SN:$USER1.FS.USER1
% 3 :10SN:$USER1.MSG.PROT
% 3 :10SN:$USER1.OUTPUT
% 3 :10SN:$USER1.PROG.C
% 3 :10SN:$USER1.SYS.SDF.LOGON.USERPROC
/start-posix-shell _____ (3)
POSIX Basishell V03.0A00 created Sep 03 1997
Copyright (C) Siemens Nixdorf Informationssysteme AG 1994
All Rights Reserved
Last login: Wed Feb 11 17:42:02 1998 on term/002 _____ (4)
$ edt .profile _____ (5)
```

- (1) Melden Sie sich in gewohnter Weise an BS2000 an.
- (2) Lassen Sie sich mit dem BS2000-Kommando /SHOW-FILE-ATTRIBUTES das Inhaltsverzeichnis Ihrer Benutzerkennung anzeigen.
- (3) Rufen Sie die POSIX-Shell mit dem BS2000-Kommando /START-POSIX-SHELL auf.
- (4) Sie sind als POSIX-Shell-Benutzer akzeptiert.
- (5) Erzeugen Sie die Datei *.profile* mit dem POSIX-Editor *edt*.  
Da die Datei noch nicht vorhanden ist, legt *edt* eine neue Datei an (siehe Seite 57).

```

1.00 alias ll='ls -l'
2.00 alias la='ls -al'
3.00 PS1="$PWD> "
4.00
5.00
6.00
7.00
8.00
9.00
10.00
11.00
12.00
13.00
14.00
15.00
16.00
17.00
18.00
19.00
20.00
21.00
22.00
POSIX editor ready for file .profile: new file
return          0000.00:001(0)
LTG      EM:1          TAST

```

```

$ . .profile _____ (6)
/home/user1> la _____ (7)
total 84
drwxr-xr-x  5 USER1    USROther   2048 Dec 22 14:03 .
drwxr-xr-x 63 SYSROOT  POSSYS     2048 Dec 22 06:35 ..
-rw-r--r--  1 USER1    USROther    48 Dec 22 14:02 .profile
-rw-----  1 USER1    USROther   2576 Dec 22 14:06 .sh_history
drwxr-xr-x  2 USER1    USROther   2048 Dec 15 17:18 c-source
drwxr-xr-x  2 USER1    USROther   8192 Dec  5 13:47 lost+found
-rw-r--r--  1 USER1    USROther    94 Dec 21 14:02 letter1
drwxr-xr-x  2 USER1    USROther   2048 Dec 19 15:05 test
...
/home/user1> cd c-source _____ (8)

```

- (6) Nach der Erstellung der *.profile*-Datei mit dem *edt* und Verlassen des Editors mit dem Kommando *return* soll die *.profile*-Datei in der aktuellen Shell ausgewertet werden. Dazu geben Sie das Punkt-Kommando *. .profile* ein.
- (7) Die POSIX-Shell meldet sich mit dem neu definierten Bereitzeichen, das den aktuellen Pfad */home/user1* ausgibt. Sie lassen sich das Inhaltsverzeichnis mit allen Dateien über das mit dem Aliasnamen *la* definierte Kommando anzeigen.
- (8) Wechseln Sie in das Unterverzeichnis *c-source*, in dem Sie beispielsweise Ihre C-Programme speichern.

```
/home/user1/c-source> bs2cp bs2:prog.c prog.c ----- (9)
/home/user1/c-source> 1a
total 60
drwxr-xr-x  2  USER1      USER1GRP  2048  Jul  6 .
drwxr-xr-x  2  USER1      USER1GRP  2048  Jul  6 ..
-rw-r--r--  1  USER1      USER1GRP  2048  Jul  6 prog.c
/home/user1/c-source> cat prog.c ----- (10)
#include <stdio.h>
main()
{
    printf("hello world\n");
    return(0);
}
/home/user1/c-source> cc -o prog prog.c ----- (11)
/home/user1/c-source> prog ----- (12)
hello world
/home/user1/c-source> exit ----- (13)
...
----- (14)
/exit-job ----- (15)
```

- (9) Kopieren Sie die im BS2000-Dateisystem liegende Datei *prog.c* in das POSIX-Dateisystem. Die Datei wird in das aktuelle Verzeichnis *c-source* geschrieben.
- (10) Lassen Sie sich den Inhalt der Datei *prog.c* mit *cat* ausgeben.
- (11) Übersetzen Sie die Datei *prog.c* mit dem C-Compiler. Das Ergebnis des Übersetzungslaufs soll in die Datei *prog* geschrieben werden.
- (12) Lassen Sie das Programm *prog* ablaufen. Es gibt die Zeichenfolge „hello world“ auf dem Bildschirm aus.
- (13) Beenden Sie mit dem Kommando *exit* die POSIX-Shell.
- (14) Eingabe von weiteren BS2000-Kommandos, falls gewünscht.
- (15) Melden Sie sich am BS2000 ab.

---

## 4 BS2000-Softwareprodukte im Umfeld von POSIX

Dieses Kapitel wendet sich an alle POSIX-Benutzer. Es gibt Ihnen einen kurzen Überblick über die Softwareprodukte im Umfeld von POSIX. Die Abschnitte 4.1 bis 4.14 behandeln BS2000-Produkte, die an POSIX angepaßt sind und mit POSIX-Dateien arbeiten können. Ab Abschnitt 4.15 auf Seite 75 werden Produkte vorgestellt, die mit POSIX ins BS2000/OSD portiert wurden. Die Menge portierten Produkten wächst ständig.

### 4.1 Binder-Lader-System ab BS2000/OSD-BC V2.0

Zum Binden und Laden ausführbarer Programme verwendet POSIX das Binder-Lader-System des BS2000/OSD, dessen wichtigste Bestandteile der Binder BINDER und der dynamische Bindelader DBL sind. Das Binder-Lader-System wird von POSIX immer dann aufgerufen, wenn ein POSIX-Benutzer ein Programm aufruft oder wenn er beim Übersetzen eines Quellprogrammes ein ausführbares Programm als Ergebnis anfordert. Eine externe Benutzerschnittstelle, wie sie BINDER im BS2000/OSD hat, ist in der POSIX-Umgebung nicht verfügbar. Für das Binden und Laden von Programmen in der POSIX-Umgebung gelten die Konventionen, die durch die Compiler und durch POSIX-übliche Programmaufrufe festgelegt sind.

Wenn in der POSIX-Umgebung Benutzerprogramme geladen werden, die noch unbefriedigte Externverweise („unresolved externals“) enthalten, werden Meldungen des Binder-Lader-Systems in der POSIX-Shell ausgegeben.

Informationen über das Binder-Lader-System des BS2000/OSD können Sie dem Handbuch „Bindelader-Starter“ [14] entnehmen.

Zum Binden in der POSIX-Shell stehen die Kommandos *cc*, *c89* und *CC* zur Verfügung. Sie sind in den Handbüchern „POSIX-Kommandos des C/C++-Compilers“ [4] und [5] ausführlich beschrieben.

## 4.2 C/C++-Compiler ab V2.2

Die BS2000-Compiler C V2.2, C++ V2.2 und C/C++ ab V3.0 können ab BS2000/OSD-BC V2.0 sowohl aus der BS2000-Umgebung (mit SDF) als auch aus der POSIX-Umgebung (POSIX-Shell) aufgerufen und mit Optionen gesteuert werden. Mit C/C++ V3.0 sind die beiden Produktlinien C und C++ zu einem Produkt C/C++ zusammengeführt.

### Compilersteuerung über die SDF-Schnittstelle

Sämtliche Compiler-Ein-/Ausgaben sind sowohl im BS2000-Dateisystem (DMS/PLAM) als auch im POSIX-Dateisystem möglich:

- Eingabe von Quellprogrammen
- Eingabe von Include-Dateien
- Ausgabe von LLMs
- Ausgabe von wiederübersetzbaren Quellprogrammen
- Ausgabe von Übersetzungslisten
- Ausgabe von Meldungslisten
- Ausgabe von CIF-Informationen

Beliebige Mischfälle, d.h. die Ein- und Ausgabe sowohl von BS2000- als auch von POSIX-Dateien in einem Übersetzungslauf, sind möglich.

Die SDF-Schnittstelle der Compiler ist in folgenden Benutzerhandbüchern beschrieben:

- „C V2.2A (BS2000/OSD), C-Compiler“
- „C++ V2.2A (BS2000/OSD), C++-Compiler“
- „C/C++ V3.0 (BS2000/OSD), C/C++-Compiler“

### Compilersteuerung über die POSIX-Shell-Schnittstelle

Für die Steuerung der C/C++-Compiler aus der POSIX-Umgebung stehen folgende POSIX-Kommandos zur Verfügung:

- |                  |  |
|------------------|--|
| <i>cc, c89</i>   | Übersetzen von C-Sourcen                               |
| <i>CC</i>        | Übersetzen von C++-Sourcen                             |
| <i>ccxref</i>    | Aufruf des globalen Listengenerators in C und C++ V2.2 |
| <i>cclistgen</i> | Aufruf des globalen Listengenerators in C/C++ ab V3.0  |

Diese Kommandos sind in den Handbüchern „POSIX-Kommandos des C/C++-Compilers“ [4] und [5] ausführlich beschrieben.

Sämtliche Compiler-Ein-/Ausgaben erfolgen ausschließlich im POSIX-Dateisystem.

In die Aufrufkommandos *cc*, *c89* und *CC* ist auch eine Binde-Phase integriert, in der die übersetzten Objekte zu einer ausführbaren Einheit gebunden werden können.

Mit den Optionen und Operanden der oben genannten Aufrufkommandos sind weitgehend die Leistungen und Funktionen abgedeckt, die mit der Compiler-Steuerung über die SDF-Schnittstelle zur Verfügung stehen. Die Syntax der POSIX-Kommandos ist an der Definition im XPG4-Standard bzw. an den in SINIX/UNIX üblichen Shell-Kommandos orientiert.

### **Hinweise zu CRTE**

Da CRTE (Common RunTime Environment) die Laufzeitumgebung für C- und C++-Programme bereitstellt, ist es Voraussetzung für die Verwendung des C/C++-Compilers.

Die Compiler C und C++ V2.2 sind mit CRTE ab V2.0 einsetzbar. Für den Einsatz von C/C++ ab V3.0 ist CRTE ab V2.1 Voraussetzung.

Mit CRTE ab V2.1 stehen alle C-Programmschnittstellen zur Verfügung, mit denen Programme nach dem XPG4.2-Standard (UNIX95) erstellt werden können.

### **Zeichensätze für Ein-/Ausgabe-Dateien**

Die Quellprogramme und Include-Dateien können im EBCDIC- oder ASCII-Code vorliegen. Somit ist es auch möglich, Quellprogramme aus Dateisystemen zu übersetzen, die auf einem fernen SINIX-/UNIX-Rechner liegen. Alle Dateien eines Dateisystems (lokales POSIX-Dateisystem oder eingehängtes fernes Dateisystem) müssen im selben Zeichensatz vorliegen, d.h. im POSIX-Dateisystem im EBCDIC-Code, in fernen SINIX/UNIX-Dateisystemen im ASCII-Code.

Die Umgebungsvariable *IO\_CONVERSION* muß mit dem Wert „YES“ belegt sein (siehe dazu auch Seite 27).

Der Ausgabe-Zeichensatz der Textdateien (Listen etc.) richtet sich nach dem Zeichensatz des Zieldateisystems.

Es wird generell EBCDIC-Ablaufcode erzeugt.

## 4.3 EDT ab V16.5

Der EDT kann im BS2000 mit /START-EDT oder aus der POSIX-Shell mit dem POSIX-Kommando *edt* aufgerufen werden.

Mit dem EDT V16.5A können Sie POSIX-Dateien erzeugen und bearbeiten. Dazu müssen das Subsystem POSIX und das Laufzeitsystem CRTE V2.0A gestartet sein.

### POSIX-Dateien mit dem EDT bearbeiten

Der EDT kann POSIX-Dateien erzeugen, einlesen, kopieren, zurückschreiben und schließen. Dazu stehen die Anweisungen @XOPEN, @XCOPY, @XWRITE und @CLOSE zur Verfügung, die im Handbuch „EDT V16.5A“ [17] ausführlich beschrieben sind.

### Konventionen für Dateinamen

Der EDT kann nur Dateinamen und Pfadnamen bis zur maximalen Länge von 256 Zeichen verarbeiten. Wenn der Pfadname länger ist, muß vorher innerhalb der POSIX-Shell mit dem Kommando *cd* in ein Unterverzeichnis positioniert werden.

Der Name einer POSIX-Datei ist folgendermaßen definiert:

xpath::=chars|str-var

Zeichenfolge mit einer maximalen Länge von 256 Zeichen.

Zeichenfolge, die den Namen einer POSIX-Datei im POSIX-Dateisystem angibt (evtl. mit Dateiverzeichnis).

Nichtabdruckbare Zeichen, Leerzeichen und andere Trennzeichen innerhalb eines Dateinamens sind nur bei Angabe einer Zeichenfolgevariablen (str-var) möglich.

Wenn der Dateiname Kleinbuchstaben enthält, müssen Sie bei Eingabe von einer Datensichtstation vorher @LOWER ON oder @PAR LOWER=ON einschalten.

Der EDT positioniert nicht innerhalb des POSIX-Dateisystems. Die Dateinamen beziehen sich immer auf das aktuelle Dateiverzeichnis, außer wenn der Dateiname mit / beginnt. In diesem Fall bezieht sich der Dateiname auf das Root-Verzeichnis.

## Satzlänge

Der EDT liest die Daten zeichenweise ein. Das Satzende wird durch das Satzende-Kennzeichen X'15' bzw. X'0A' (abhängig vom Code) erkannt.

Die erlaubte Satzlänge liegt zwischen 1 und 256 Zeichen.

Zeichenketten, die länger als 256 Zeichen sind, werden ab dem 257. Zeichen abgeschnitten. Der EDT gibt dann folgende Fehlermeldung aus:

```
% EDT1253 (SOME) RECORD(S) TRUNCATED
```

Zeichenketten der Länge 0 können im EDT-Datenbereich nicht dargestellt werden; sie müssen extra behandelt werden. Beim Einlesen wird, abhängig vom AUTOFORM-Modus (siehe EDT-Anweisung @BLOCK), folgendermaßen vorgegangen:

- AUTOFORM-Modus ausgeschaltet:  
Zeichenketten der Länge 0 werden ignoriert. Es wird kein Satz angelegt.
- AUTOFORM-Modus eingeschaltet:  
Leerzeilen erhalten als Inhalt das Zeilenende-Kennzeichen X'0D'; sie werden im Datenbereich angelegt.

Analog dazu wird beim Schreiben einer Datenzeile mit dem Inhalt X'0D' der AUTOFORM-Modus ausgewertet:

- AUTOFORM-Modus ausgeschaltet:  
Datenzeilen mit dem Inhalt X'0D' als solche in die POSIX-Datei geschrieben.
- AUTOFORM-Modus eingeschaltet:  
Datenzeilen mit dem Inhalt X'0D' als Satz der Länge 0 in die POSIX-Datei geschrieben.

## Verarbeiten von Daten im ASCII-Code

Sie müssen dem EDT mit dem Operanden CODE mitteilen, ob die Daten im ASCII-Code vorliegen bzw. ob sie im ASCII-Code in die POSIX-Datei abgelegt werden sollen.

Es wird eine feste Umwandlungstabelle verwendet. Die Tabelle entspricht der Zuordnung von EDF03IRV zu ISO646 internationaler 7-Bit-Code (gleichwertig zur Zuordnung EDF041 zu ISO8859-1).

Die Daten in der EDT-Arbeitsdatei können Sie im ASCII-Code hexadezimal mit @PAR HEX=ON und der Voreinstellung @PAR CODE=ISO anzeigen und im Datenfenster verändern.

Im L-Modus können Sie Daten im ASCII-Code hexadezimal durch die Voreinstellung @INPUT HEX ISO eingeben.

Wenn Sie den EDT aus der POSIX-Shell aus aufrufen, können Sie bereits beim Aufruf mit der Option -k angeben, daß der Inhalt der Datei vor dem Editieren von ASCII nach EBCDIC konvertiert und vor dem Zurückschreiben wieder nach ASCII zurückkonvertiert werden soll.

## 4.4 File-Transfer-Produkte (FT-BS2000, FTOS-BS2000, *openFT*)

FT-BS2000 V5.2 und FTOS-BS2000 V3.0 unterstützen für die Funktionen Dateitransfer und Dateiverwaltung den Zugriff auf POSIX-Dateien.

Ab Version 6 erhielten die File-Transfer-Produkte die neue Produktbezeichnung *openFT*. *openFT* (BS2000) V6.0 löst FT-BS2000 V5.2 ab. Optionale Komponenten sind *openFT-OS* (BS2000) zur Realisierung der FTAM-Funktionalität und *openFT-AC* (BS2000) für den mit der FTAC-Funktionalität gebotenen Zugangs- und Zugriffsschutz.

Das POSIX-Dateisystem wird ab BS2000/OSD-BC V2.0 vom Subsystem POSIX bereitgestellt. Dieses Subsystem muß gestartet sein, damit die POSIX-Funktionen von FT-BS2000, FTOS-BS2000 und *openFT* (BS2000) genutzt werden können. Außerdem muß POSPRRTS gestartet sein.

### Angabe einer POSIX-Datei

Eine POSIX-Datei muß in den FT-Kommandos durch eine besondere Syntax angegeben werden: Dateinamen, die mit / oder ./ beginnen, werden als voll- bzw. teilqualifizierte POSIX-Dateinamen interpretiert. Dateien, die nicht mit einem dieser Zeichen beginnen, gelten als BS2000-Dateien.

Nähere Informationen können Sie dem Handbuch „*openFT* (BS2000)“ [19] entnehmen.

## 4.5 HSMS / HSMS-SV ab V2.0B

HSMS V2.0B ermöglicht auch das Sichern, Archivieren und Restaurieren von Dateien, die auf fernen Rechnern im Netz liegen. Das zu bearbeitende Dateisystem kann entweder das lokale BS2000-UFS sein oder ein fernes UNIX-Dateisystem, das der BS2000-Systemverwalter am Dateiverzeichnis „HSMS“ des lokalen BS2000-UFS einhängen muß. Das Dateiverzeichnis „HSMS“ befindet sich direkt unter dem Root-Verzeichnis „/“, das in jedem UNIX-Dateisystem vorhanden ist und als zentraler Einstieg dient.

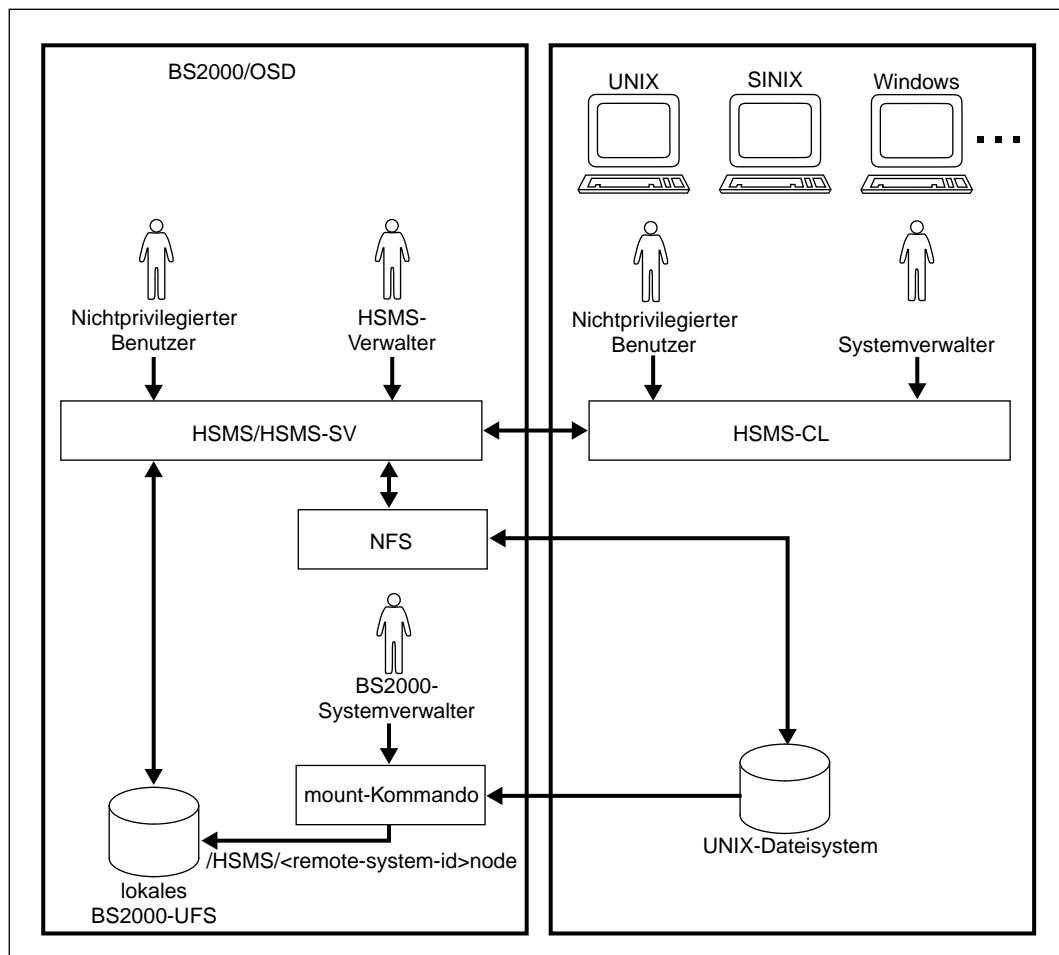


Bild 22: Ferne UNIX-Dateisysteme am Dateiverzeichnis /HSMS

Das lokale BS2000-UFS wird - abhängig von der eingesetzten BS2000-Version - unterschiedlich erstellt und verwaltet:

- Bei BS2000/OSD-BC V1.0 wird das BS2000-UFS beim First-Startup des Subsystems NFS erstellt.
- Ab BS2000/OSD-BC V2.0 ist das lokale BS2000-UFS das POSIX-Dateisystem.

Die folgenden HSMS-Anweisungen stehen dem HSMS-Verwalter und - in eingeschränktem Umfang - auch dem nichtprivilegierten Benutzer zur Verfügung, um Dateien eines UNIX-Dateisystems bearbeiten zu können:

- **ARCHIVE-NODE-FILES**  
Knotendateien archivieren
- **BACKUP-NODE-FILES**  
Knotendateien sichern
- **COPY-NODE-SAVE-FILE**  
Knoten-Sicherungsdatei kopieren
- **MODIFY-NODE-PARAMETERS**  
Parameter eines Knoten-S0 ändern
- **RESTORE-NODE-FILES**  
Knotendateien restaurieren
- **SELECT-NODE-FILES**  
Knotendateien auswählen
- **SHOW-NODE-PARAMETERS**  
Parameter von Knoten-S0 ausgeben

Nähere Informationen können Sie dem Handbuch „HSMS / HSMS-SV“ [22] entnehmen.

## 4.6 NFS ab V1.2

Um mit den Dateisystemen eines fernen Rechners arbeiten zu können, muß das Softwareprodukt NFS (Network File System) auf dem fernen und lokalen Rechner installiert sein. Am fernen Rechner (NFS-Server) muß das einzuhängende Dateisystem mit dem NFS-Kommando *share* bereitgestellt und am lokalen Rechner (NFS-Client) mit dem NFS-Kommando *mount* eingehängt werden. Danach kann auf das ferne Dateisystem vom lokalen Rechner aus zugegriffen werden.

NFS V1.2 enthält nur die NFS-Verwaltungsfunktionen. Die übrigen Kommandos, die in NFS V1.0 enthalten sind, sind jetzt Bestandteil von POSIX-BC. Deshalb ist POSIX-BC Voraussetzung für NFS V1.2.

Nähere Informationen können Sie dem Handbuch „NFS V1.2A“ [12] entnehmen.

## 4.7 SDF-A ab V4.0

SDF-A V4.0 unterstützt die Angabe von POSIX-Pfadnamen und POSIX-Dateinamen. Die Datentypen *<posix-filename>* und *<posix-pathname>* wurden mit SDF-A V4.0 eingeführt.

Nähere Informationen können Sie dem Handbuch „SDF-A“ [26] entnehmen.

## 4.8 SECOS ab V2.1

POSIX verwendet für die Verwaltung und Zugangskontrolle der POSIX-Benutzer den Baustein SRPM des Softwareprodukts SECOS.

Wenn SECOS in Ihrem System nicht installiert ist, so ist doch der für POSIX relevante Teil von SRPM im Grundausbau des BS2000 enthalten.

Nähere Informationen zur BS2000-Verwaltung der POSIX-Benutzer finden Sie in diesem Handbuch im Kapitel „POSIX-Benutzer verwalten“ auf Seite 113.

Die Zugangskontrolle für Anwender, die sich von einem UNIX- oder SINIX-Rechner mit dem Kommando *rlogin* an einen BS2000-Rechner anschließen wollen, ist im Abschnitt „Zugang zur POSIX-Shell“ auf Seite 48 beschrieben.

Wenn SECOS im Einsatz ist, stehen Ihnen für POSIX die folgenden zusätzlichen Möglichkeiten zur Verfügung:

- Verwendung des Privilegs POSIX-ADMINISTRATION für ausgewählte Benutzerkennungen (SRPM).
- Protokollierung und Auswertung sicherheitsrelevanter Ereignisse, die POSIX betreffen, mit SAT.

Neben den allgemeinen Möglichkeiten der Überwachung von Benutzerkennungen, DVS-Dateiobjekten und Ereignissen sind folgende Ereignisse speziell für POSIX definiert:

- Ereignis JFK: POSIX-Task erzeugen
- Ereignis UPA: Kommando /MODIFY-POSIX-USER-ATTRIBUTES
- Ereignis UPD: Kommando /MODIFY-POSIX-USER-DEFAULTS

Die sicherheitsrelevanten Ereignisse der Privilegienverwaltung - z. B. das Privileg POSIX-ADMINISTRATION vergeben - werden stets mit SAT protokolliert.

Nähere Informationen können Sie dem Handbuch „SECOS“ [13] entnehmen.

## 4.9 SOCKETS/XTI (POSIX-SOCKETS) ab V1.1

Mit dem Subsystem POSIX stehen auch die SOCKETS/XTI-Schnittstellen zur Verfügung. Es handelt sich dabei um Schnittstellen zur Programmierung von Netzfunktionen, mit denen der Zugang zum Internet über TCP/IP und UDP/IP ermöglicht wird. Diese Schnittstellen gewährleisten damit den Zugang zur offenen Netzwerk-Welt.

Die SOCKETS/XTI-Schnittstellen werden mit POSIX-SOCKETS ausgeliefert und sind in einer eigenen Bibliothek definiert. Wenn diese Bibliothek in eine POSIX-Anwendung eingebunden ist, stellen die SOCKETS/XTI-Schnittstellen über das Subsystem POSIX und das Transportsystem BCAM die Verbindung zum Netzwerk her.

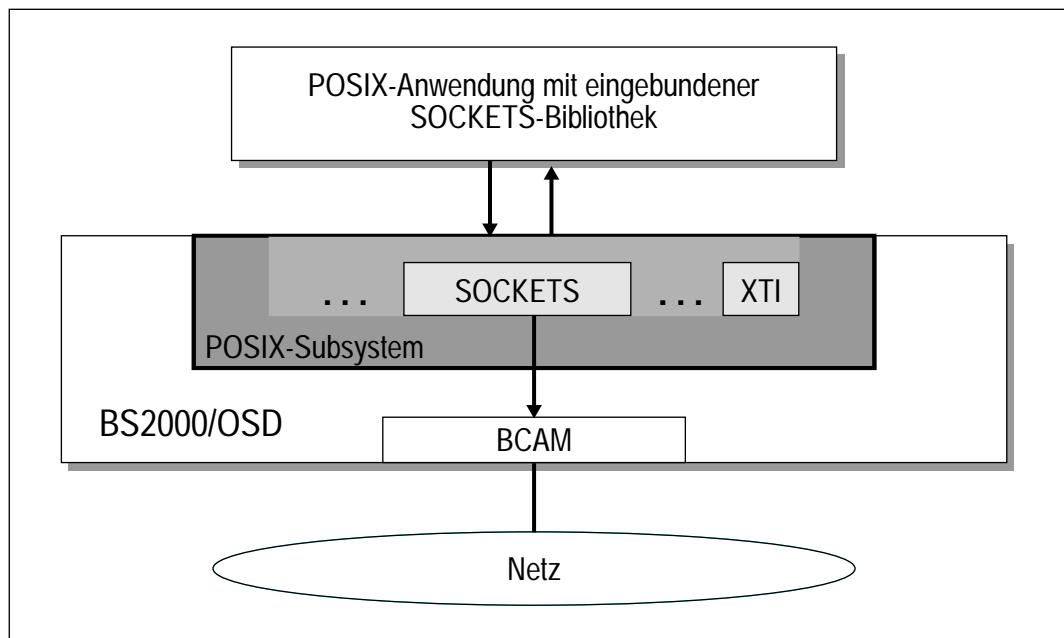


Bild 23: SOCKETS/XTI im BS2000/OSD und in POSIX

Nähere Informationen können Sie dem Handbuch „SOCKETS/XTI für POSIX“ [3] entnehmen.

## 4.10 SPOOL ab V3.0

Mit SPOOL ab V3.0 können Sie POSIX-Dateien mit BS2000- oder POSIX-Kommandos ausdrucken.

### *Beispiel*

Drucken der POSIX-Datei `/home/psxroot/usr1` aus dem BS2000:

```
/PRINT-DOCUMENT FROM-FILE=' /home/psxroot/usr1',...
```

Drucken derselben POSIX-Datei aus der POSIX-Shell:

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

```
.
```

```
.
```

```
/home/psxroot> lp usr1
```

```
/home/psxroot> exit
```

POSIX-Dateien können Sie mit dem POSIX-Kommando `lp` auf dem angeschlossenen BS2000-Drucker ausdrucken. Das Kommando `lp` verwendet den BS2000-SPOOL zum Ausdrucken. Es wird keine ID für einen Druckauftrag vergeben.

Die Druckaufträge können nur über den BS2000-SPOOL verwaltet werden.

Nähere Informationen können Sie den Handbüchern zu „SPOOL“ ([31] und [32]) entnehmen.

## 4.11 TLI (POSIX-NSL) ab V1.0

Neben den SOCKETS-Schnittstellen (siehe Seite 69) stehen optional auch die TLI-Netzwerk-Schnittstellen zur Verfügung. Sie ermöglichen ebenfalls den Zugriff auf das Internet auf Basis von TCP/IP und UDP/IP. Die TLI-Schnittstellen werden mit POSIX-NSL ab V1.0 ausgeliefert.

Die TLI-Schnittstellen bestehen - ebenso wie die SOCKETS-Schnittstellen - aus einer Reihe von Bibliotheksfunktionen, die über das Transportsystem BCAM die Verbindung zum Netzwerk herstellen.

## 4.12 AID ab V2.2

Mit der Dialog-Testhilfe AID ab V2.2A können Sie neben reinen BS2000-Programmen auch reine POSIX-Programme und gemischte Programme testen. Gemischte Programme benutzen sowohl BS2000-Programmschnittstellen als auch POSIX-Programmschnittstellen. Das Testen wird ermöglicht durch Erweiterungen in den AID-Kommandos %AID und %STOP ab BS2000/OSD-BC V3.0, sowie durch das POSIX-Kommando *debug*.

Das AID-Kommando %AID wurde um zwei neue Operanden FORK={OFF | NEXT | ALL} und EXEC={OFF | ON} ergänzt. Diese bewirken, daß unmittelbar nach einem *fork()*- bzw. *exec()*-Aufruf das Programm unterbrochen wird und in den Testmodus wechselt, so daß Sie wie gewohnt AID-Kommandos zum Testen Ihres Programms eingeben können.

Das AID-Kommando %STOP wurde ebenfalls um zwei neue Operanden T=tsn (Task Sequence Number) und PID=pid (Process Identification) erweitert, über die Sie eine durch *fork()* entstandene Task unterbrechen können.

AID meldet sich mit der Prozeßnummer (pid) der unterbrochenen Task, und Sie können den weiteren Verlauf dieser Task über AID-Kommandos kontrollieren.

Das POSIX-Kommando *debug* ermöglicht es Ihnen, in der POSIX-Shell ein Programm mit LSD zu laden oder einen bereits laufenden Prozeß zu unterbrechen und in den Testmodus zu versetzen:

`debug programe`

Das Programm wird mit LSD in einer Fork-Task geladen und in den Testmodus versetzt; Sie können AID-Kommandos eingeben. Das Kommando 'debug programe' in der POSIX-Shell entspricht somit dem BS2000-Kommando

`LOAD-PROGRAM programe, ... TEST-OPTIONS=*YES`  
in der BS2000-Umgebung.

`debug -p pid`

Der Prozeß mit der angegebenen pid wird von AID übernommen und unterbrochen. 'debug -p pid' in der POSIX-Shell entspricht dem oben erwähnten AID-Kommando %STOP PID=pid, das Sie im BS2000-Systemmodus oder im Testmodus einer Task eingeben können.

Dumps von gemischten oder POSIX-Programmen werden wie bisher im BS2000 abgelegt und können dort bearbeitet werden. Falls AID zum Dump eines POSIX-Programms die LSD über das AID-Kommando %SYMLIB nachladen soll, müssen Sie beachten, daß %SYMLIB nicht auf POSIX-Dateien zugreifen kann. Die entsprechende Datei muß zunächst mit dem POSIX-Kommando *bs2cp* als L-Element in eine PLAM-Bibliothek im BS2000 kopiert werden und kann dann mit %SYMLIB zugewiesen werden.

Wie POSIX-Programme und gemischte Programme mit AID getestet werden, ist ausführlich in [34] bzw. [35] beschrieben.

## 4.13 SORT ab V7.5

Ab BS2000/OSD-BC V2.0 kann SORT ab V7.5A auch POSIX-Dateien verarbeiten.

POSIX-Dateien können in der SORT-Steueranweisung ASSIGN-FILES oder im Kommando SORT-FILE als Eingabedatei (Operand INPUT-FILES) oder als Ausgabedatei (Operand OUTPUT-FILES) zugewiesen werden.

Zur Unterscheidung von BS2000-Dateinamen müssen POSIX-Dateinamen in den Operanden INPUT-FILES und OUTPUT-FILE in Hochkommata angegeben werden.

Arbeits- und Hilfsdateien dürfen keine POSIX-Dateien sein.

Die Daten in den POSIX-Dateien liegen im Text-Format vor, das von SORT nicht unmittelbar verarbeitet werden kann. Sie werden daher vor der Bearbeitung durch die Sortierroutinen von SORT in Sätze variabler Länge umgewandelt, denen jeweils ein Satzlängenfeld vorangestellt wird.

Nach dem Sortiervorgang wandelt SORT die sortierte Ausgabedatei wieder in das Text-Format um, wenn sie im POSIX-Dateisystem gespeichert werden sollen.

Die interne Verwendung von Sätzen variabler Satzlänge bewirkt, daß sich die Position der Benutzerdaten im Satz um das Satzlängenfeld verschiebt. Für den Anwender von POSIX-Dateien hat das im Normalfall jedoch keine Auswirkungen. SORT berechnet die Feldpositionen bei Sätzen aus POSIX-Dateien standardmäßig relativ zum Anfang der Benutzerdaten.

Will der Anwender dennoch auf das interne Satzlängenfeld zugreifen, z.B. um die Sätze nach ihrer Länge zu sortieren, steht der Operand IGNORE-LENGTH-FIELD in der Anweisung SET-SORT-OPTIONS und im Kommando SORT-FILE zur Verfügung.

Die Angabe IGNORE-LENGTH-FIELD=\*NO bewirkt, daß sowohl bei variabel langen Sätzen in BS2000-Dateien, als auch bei Sätzen in POSIX-Dateien die Positionen innerhalb des Satzes ab Satzanfang berechnet werden. Damit beginnen die Benutzerdaten an Position 5 im Satz.

Die Verschlüsselung des Satzende-Kennzeichens wird durch den Operanden CODE in der Anweisung ASSIGN-FILES und im Kommando SORT-FILE bestimmt. Bei CODE=EBCDIC wird das Satzende-Kennzeichen als X'0A' verschlüsselt, bei CODE=ASCII als X'15'.

Bei der Verwendung von POSIX-Dateien als Ausgabedatei ist darauf zu achten, daß die Ausgabesätze keine Zeichen enthalten, die als Satzende-Kennzeichen interpretiert werden. Das bedeutet im einzelnen:

- In der Anweisung SORT-RECORDS oder im Kommando SORT-FILE dürfen keine Konstantenfelder angegeben werden, die Satzende-Kennzeichen enthalten.
- Die Sätze einer BS2000-Eingabedatei dürfen keine Satzende-Kennzeichen enthalten, wenn die Ausgabedatei eine POSIX-Datei sein soll.
- Die Sortierart „Adresslistensortieren“ darf nicht verwendet werden, da nicht sichergestellt werden kann, daß die Adreßfelder keine als Satzende-Kennzeichen interpretierbaren Zeichen enthalten.



Das Kommando *sort*, das in einer POSIX-Shell aufgerufen werden kann, ist nicht identisch mit einem Aufruf des Produkts SORT.

Nähere Informationen können Sie dem Handbuch „SORT“ [33] entnehmen.

## 4.14 TCP-IP-AP ab V2.0

Mit TCP-IP-AP ab V2.0 kann von fernen Workstations (UNIX-/SINIX-Rechner) oder PCs (Windows, Windows 95, Windows NT) mit *ftp* auf POSIX-Dateiverzeichnisse zugegriffen werden. Voraussetzung dafür ist, daß auf dem BS2000/OSD-System der *ftp*-Server-Task mit dem Kommando START-FTP-DEMON gestartet wurde.

Mit der Pfadangabe %POSIX bei *cd* oder *lcd* in einer *ftp*-Sitzung kann in das POSIX-UFS gewechselt werden.

### Beispiel

Das folgende Beispiel zeigt einen Ausschnitt einer *ftp*-Sitzung:

```
$ ftp bs2server _____ (1)
Connected to BS2SERVER.
220 bs2server FTP server (Version V02.2A00, Oct 13 1997 09:32:49) ready.
User (BS2SERVER:(none)): user1 _____ (2)
331 Password required for user1.
Password: _____
332 Account required.
Account: m0815xyz _____ (2)
230 User USER1 logged in.
Ftp> cd %POSIX _____ (3)
200 "/home/user1" is current directory now
Ftp> ...
Ftp> bye _____ (4)
221 Goodbye.
```

- (1) Geben Sie beim Kommando *ftp* den BS2000/OSD-Server an, auf dem das POSIX-Dateisystem installiert ist.
- (2) Geben Sie Benutzerkennung, Paßwort und Account Ihrer BS2000-Kennung auf diesem Server ein.
- (3) Die Angabe %POSIX bei *cd* bewirkt, daß Sie vom BS2000-Dateisystem in das POSIX-Dateisystem wechseln. Danach befinden Sie sich in dem HOME-Verzeichnis der POSIX-Kennung, die Ihrer BS2000-Kennung zugeordnet ist.
- (4) Mit dem *ftp*-Befehl *bye* verlassen Sie das POSIX-Dateisystem, melden sich bei BS2000 ab und schließen die *ftp*-Sitzung.

Nähere Informationen können Sie dem Handbuch „TCP-IP-AP“ [40] entnehmen.

## 4.15 TCP-IP-SV

Das Softwareprodukte TCP-IP-SV stellt folgende TCP/IP-Services bereit:

- Domain Name Service (DNS) Resolver
- Bootstrap-Protokoll (BOOTP) Server
- Dynamic Host Configuration Protocol (DHCP) Server
- Trivial File Transfer Protocol (TFTP) Server

Damit werden Funktionen zur Verfügung gestellt, die die Interoperabilität von Kommunikationspartnern in 'offenen' TCP/IP basierten Netzen ermöglicht.

Die Produkte der Liefereinheit TCP-IP-SV sind Portierungen von entsprechenden Internet-Standardprodukten aus der 'offenen' Welt, die jedoch den spezifischen Gegebenheiten, die in BS2000/OSD vorhanden sind, angepaßt wurden. Damit ist eine einheitliche Benutzerschnittstelle, die Administration und die Interoperabilität der Produkte über Systemgrenzen hinweg gewährleistet.

Näheres dazu erfahren Sie im Handbuch „TCP-IP-SV“ [41].

## 4.16 DCE (BS2000)

Unter dem Titel DCE (Distributed Computing Environment) legte „The Open Group“ (Nachfolgeorganisation der Open Software Foundation OSF und X/Open) eine herstellerunabhängige Umgebung für die Entwicklung und den Ablauf von verteilten Anwendungen fest. DCE (BS2000) V1.0 ist eine Portierung von DCE (SINIX) und basiert auf OSF DCE V1.02. Mit DCE (BS2000) bietet Siemens Nixdorf das DCE-Paket durchgängig auf allen Serverplattformen (BS2000, Reliant UNIX, Windows NT) an.

DCE umfaßt zur Realisierung von Client-Server-Anwendungen verschiedene Funktionen und Services. Zu den Grundbestandteilen gehört:

- Remote Procedure Call (RPC)  
Basis-Mechanismen zur effektiven Erstellung von verteilten Anwendungen
- Threads  
zur Parallelisierung von Programmabläufen innerhalb von Prozessen
- Directory Services (auch Naming Services genannt)  
ermöglichen dem Benutzer den Zugriff auf Betriebsmittel im Netz über logische Namen
- Security Services  
zum sicheren Betrieb von verteilten Anwendungen in Netzen
- Time Service  
zur Synchronisation der Uhren vernetzter Systeme

- Distributed File System (DFS)  
stellt ein verteiltes Dateisystem zur Verfügung. DFS ist eine komplexe Anwendung, die auf sämtlichen anderen Diensten von DCE aufsetzt.  
Nähtere Informationen finden Sie in den Handbüchern zu „DCE (BS2000)“ ([8] bis [11]).

## 4.17 TransView SNMP-Basic-Agent BS2000

Der SNMP-Basis-Agent im BS2000 dient zum direkten Anschluß von BS2000/OSD-Systemen an ein zentrales Management-System für heterogene Netze über das De-facto-Standardprotokoll SNMP (Simple Network Management Protocol).

Die BS2000/OSD-Systeme in großen, herstellerübergreifenden Netzen aus Systemen verschiedenster Art können damit von einer Management-Plattform (von einem Leitstand) aus zentral überwacht und gesteuert werden.

Das Basisprodukt TransView SNMP-Basic-Agent BS2000 enthält:

- den Masteragenten, der das SNMP-Protokoll realisiert und damit die Grundvoraussetzung für das SNMP-Management im BS2000 ist.
- eine Schnittstelle für den Anschluß von Subagenten, die den Zugriff auf die Systeminformationen realisieren, wobei auch kundeneigene Subagenten anschließbar sind.
- den Subagenten „Application Monitor“, zur Überwachung von Benutzeranwendungen, BCAM-Anwendungen, Subsystemen des BS2000 und Protokolldateien.
- den Subagenten und die Management-Anwendung für den „Konsol-Monitor“ zur Darstellung von BS2000-Konsolmeldungen mit der Möglichkeit der Filterung, zur Eingabe von Konsol-Kommandos und zur Beantwortung von Fragen, die auf der Konsole erscheinen.
- die Unterstützung der systemspezifischen Teile der MIB-II (System-Group und SNMP-Group entsprechend RFC 1213) für das Netzmanagement.

Nähtere Informationen finden Sie im Handbuch „TransView SNMP Basic Agent BS2000 / TransView SNMP Standard Collection BS2000“ [42].

# 5 POSIX installieren

Dieses Kapitel wendet sich an die Systemverwalter von BS2000 und POSIX, die POSIX und weitere POSIX-Programmpakete installieren wollen. Es informiert Sie über

- den Lieferumfang von POSIX
- die Installationsschritte für POSIX
- das Installationsprogramm von POSIX
- die Informationsdatei von POSIX

## 5.1 Lieferumfang

Die grundlegenden POSIX-Funktionen und -Kommandos sind Bestandteil des BS2000-Grundausbaus BS2000/OSD-BC. POSIX besteht aus den Release Units:

- POSIX-BC (Subsystem POSIX und Basis-Shell)
- POSPRRTS (Laufzeitsystem für den privilegierten Teil von POSIX)
- POSIX-HEADER (Include-Header für die POSIX-Bibliotheksfunktionen)
- POSIX-SOCKETS (SOCKETS/XTI-Netzwerkschnittstellen)
- POSIX-NSL (TLI-Netzwerkschnittstellen)
- POSIX-ADDON-LIB (Schnittstellen, die nicht zum XPG4-Standard gehören)

Die POSIX-Shell wird als eigene Release-Unit POSIX-SH freigegeben.

Mit POSIX-SH ab V1.1A steht der vollständige Kommandoumfang gemäß den XPG4-Standard zur Verfügung.

Die Programmier- und Anwendungsschnittstellen (Module) für POSIX werden als Bibliotheksfunktionen für die Programmiersprache C/C++ im Rahmen des Softwareprodukts CRTE freigegeben und installiert.

Die Dateinamen der einzelnen Lieferbestandteile entnehmen Sie bitte den Freigabemitteilungen (SYSFGM) zu POSIX-BC und POSIX-SH.

## 5.2 Installationsschritte

Im Rahmen der Installation von POSIX mit IMON/SOLIS2 müssen Sie die folgenden Schritte unter der Benutzerkennung TSOS im BS2000 durchführen.

### Subsystemkatalog prüfen

Das SSC-Objekt SYSSSC.POSIX-BC.<version> (Subsystem POSIX) wird bei der Installation automatisch in den statischen Subsystemkatalog übernommen.

Nähere Informationen zum Subsystemkatalog finden Sie im Handbuch „Einführung in die Systembetreuung“ [18]. Das BS2000 muß nach dem Eintragen neu gestartet werden.

### Benötigte Subsysteme starten

Vor dem Start des Subsystems POSIX müssen Sie die Subsysteme CRTEBASY und EDT starten. Ab EDT V16.5 müssen Sie zuerst das Subsystem EDTCON und anschließend das Subsystem EDT starten.

BCAM muß ebenfalls gestartet sein, damit der Hostname im Subsystem POSIX bekannt ist.

### Benötigte Dateien mehrfach benutzbar machen

Folgende Dateien müssen Sie, falls nicht mit SOLIS2 installiert wird, mit dem Kommando /MODIFY-FILE-ATTRIBUTES mehrfach benutzbar machen (USER-ACCESS=\*ALL-USERS und ACCESS=\*>READ):

- SINENT.POSIX-BC.<version>
- SINLIB.POSIX-BC.<version>
- SINLIB.POSIX-BC.<version>.HTTPD
- SINLIB.POSIX-BC.<version>.INET
- SINLIB.POSIX-BC.<version>.ROOT
- SINLIB.POSIX-BC.<version>.SHELL
- SINLNK.POSIX-BC.<version>
- SINPRC.POSIX-BC.<version>
- SYSENT.POSIX-BC.<version>
- SYSLNK.POSIX-BC.<version>.INIT
- SYSLNK.POSIX-BC.<version>.SHELL

## Init-Prozeß vorbereiten

Beim Starten des Subsystems POSIX wird ein Init-Prozeß angestoßen, der als BS2000-Benutzerprozeß abläuft. Die Lebensdauer dieses Init-Prozesses entspricht der Lebensdauer des Subsystems POSIX. Deshalb müssen Sie vor dem Start des Subsystems POSIX das CPU-Limit der Systemkennung SYSROOT entfernen (Attribut NO\_CPU\_LIMIT=YES). Für die Standard-Batchjob-Klasse der Kennung SYSROOT muß ebenfalls das Attribut NO\_CPU\_LIMIT auf YES gesetzt werden. Die Kennungen SYSROOT und SYSPRIV müssen Sie entsperren:

```
/UNLOCK-USER USER-ID=SYSROOT  
/UNLOCK-USER USER-ID=SYSPRIV
```

Außerdem müssen Sie an SYSROOT das Privileg STD-PROCESSING vergeben:

```
/SET-LOGON-PARAMETERS USER-ID=SYSPRIV,ACCOUNT=SYSACC  
/SET-PRIVILEGE PRIVILEGE=STD-PROCESSING,USER-ID=SYSROOT
```

## Erstinstallation von POSIX auf dem POSIX-Dateisystem

Ist POSIX auf Ihrem System noch nicht installiert, so müssen Sie mit dem POSIX-Installationsprogramm unter TSOS eine „Erstinstallation“ für POSIX auf dem POSIX-Dateisystem durchführen (siehe Abschnitt „Erstinstallation“ auf Seite 85). Für die Installation von POSIX und seiner Programmpakete ist das Privileg SUBSYSTEM-MANAGEMENT erforderlich.

Die für die Installation von POSIX und seiner Programmpakete nötige root-Berechtigung (Benutzernummer 0, Gruppennummer 0) wird der Benutzerkennung TSOS beim Ablauf der Erstinstallation automatisch zugewiesen.

## Voraussetzungen für die Nutzung von CRTE ab V2.0 schaffen

Der Subsystemkatalog muß folgende SSC-Elemente der CRTE-Lieferung enthalten (version ≥ 020):

- SYSSSC.CRTE.<version>.C bzw. SYSSSC.CRTE.<version>.C.HIGH
- SYSSSC.CRTE.<version>.COM

Die Subsysteme CRTEC und CRTECOM müssen gestartet sein. Das Subsystem ILCS darf nicht gestartet sein.

Wenn Sie C/C++-Programme in der POSIX-Shell entwickeln wollen (POSIX-Kommandos *cc*, *c89* oder *CC*), müssen Sie mit dem POSIX-Installationsprogramm, Maske „Install packages on POSIX“ die Elemente der Bibliothek SINLIB.CRTE.<version> unter TSOS im POSIX-Dateisystem installieren. Anschließend müssen Sie auf die gleiche Weise die Elemente der Bibliothek SINLIB.POSIX-HEADER.<version> installieren.

### POSIX-Shell bereitstellen

Das Bereitstellen der Basis-Shell erfolgt implizit während der POSIX-Installation.

Die erweiterte POSIX-Shell (Bibliothek SINLIB.POSIX-SH.<version>) wird mit dem POSIX-Installationsprogramm, Maske „Install packages on POSIX“ unter TSOS im POSIX-Dateisystem installiert.

Zum Aufruf der POSIX-Shell müssen noch folgende Voraussetzungen erfüllt sein:

- Die Lieferdateien für die POSIX-Shell (siehe Abschnitt „Lieferumfang“ auf Seite 77) müssen mehrfach benutzbar unter der Standardbenutzerkennung des Systems (DEFLUID) stehen.
- Die Syntaxdatei SYSSDF.POSIX-SH.<version> muß mit dem Kommando /MODIFY-SDF-PARAMETERS SYNTAX-FILE-TYPE=\*SUBSYSTEM( . . . ) aktiviert werden.

Basis- und POSIX-Shell werden mit dem BS2000-Kommando /START-POSIX-SHELL (siehe Seite 157) aufgerufen und mit dem Shell-Kommando *exit* beendet.

## 5.3 POSIX-Installationsprogramm

Nach der Installation von POSIX im BS2000 müssen Sie noch POSIX und die gewünschte Software über das POSIX-Installationsprogramm unter der Benutzerkennung TSOS im POSIX-Dateisystem installieren und nötigenfalls Dateisysteme erzeugen und bearbeiten.

Dazu benötigen Sie als BS2000-Systemverwalter zusätzlich die root-Berechtigung (Benutzernummer 0, Gruppennummer 0), die bei der Erstinstallation von POSIX automatisch an TSOS vergeben wird. Außerdem muß die POSIX-Informationsdatei SYSSI.POSIX-BC.<version> schreibbar sein (ACCESS=\*WRITE).

Die Installationen können Sie im Dialog oder automatisiert durchführen:

- Im Dialog werden die Zweige des POSIX-Installationsprogramms über Masken durchlaufen. Jeder Zweig wird über individuell gestaltete Masken gesteuert und mit Daten versorgt. Eventuelle Fehleingaben und Inkonsistenzen werden im Dialog gemeldet; Korrekturen sind umgehend möglich.
- Bei automatisiertem Ablauf erfolgt die Steuerung und Dateneingabe über Parameterdateien, deren Layout genau festgelegt ist. Fehlerhafte Parameterdateien führen zum Abbruch des POSIX-Installationsprogramms.

### 5.3.1 Liefer- und Installationsverfahren für POSIX-Programmpakete

Programmpakete zur Installation unter POSIX werden für die von der Siemens Nixdorf Informationssysteme AG gelieferte Software als BS2000-PLAM-Bibliotheken mit dem Standardnamen SINLIB.<product>.<version> geliefert und unter einer beliebig wählbaren Ablagekennung eingespielt.

Diese Bibliotheken enthalten auch Kommandoprozeduren zum Installieren und Löschen der Programmpakete im POSIX-Dateisystem. Die Installation im POSIX-Dateisystem erfolgt mit der Funktion „Install packages on POSIX (Programmpakete hinzufügen)“ im POSIX-Installationsprogramm.

Derzeit liefern folgende Release-Units eine SINLIB.<product>.<version> zur Installation unter POSIX:

- POSIX-HEADER (Include-Header für die POSIX-Bibliotheksfunktionen)
- POSIX-SOCKETS (Schnittstelle zur Programmierung von Netzfunktionen; enthält SOCKETS/XTI)
- POSIX-ADDON-LIB (Schnittstellen, die nicht zum XPG4-Standard gehören)
- POSIX-NSL (TLI Netzwerkschnittstellen)
- POSIX-SH (erweiterte POSIX-Shell)
- CRTE (Common RunTime Environment)
- C (C-Compiler)
- CPP (C++-Compiler V2.2 und C/C++-Compiler ab V3.0)
- DCE-DE (Schnittstellen für verteilte Anwendungen)
- NFS (Network File System)
- COBOL85 (COBOL85-Compiler)
- TV-SBA-BS2, TV-SSA-SM2, TV-SSA-UTM, TV-SSC-BS2 (TransView SNMP)
- TCP-IP-SV (TCP/IP-Services)

Weitere Produkte sind geplant.

Programmpakete können wahlweise aus dem Software Configuration Inventory (d.h. nach dem offiziellen Lieferverfahren mit IMON-Support) oder von einer beliebigen Ablagekennung installiert werden. Das gilt auch für private Programmpakete (z.B. private POSIX-Anwendungen), da auch diese im Software Configuration Inventory registriert werden können. Auf POSIX kann immer nur eine Version bzw. ein Korrekturstand eines Produktes installiert sein.

## Installation aus dem Software Configuration Inventory (SCI)

Bei der POSIX-Installation aus dem SCI gibt es 3 verschiedene Arten der Nutzung:

1. Installation des von IMON voreingestellten Korrekturstandes des Produktes

In diesem Fall gibt der POSIX-Verwalter weder Version noch Korrekturstand an. Es wird entweder der höchste Korrekturstand des Produktes oder - falls vor dem Start des POSIX-Installationstools ein Kommando /SELECT-PRODUCT-VERSION eingegeben wurde - der durch dieses Kommando ausgewählte Korrekturstand auf POSIX installiert.

2. Installation mit frei gewähltem Korrekturstand des Produktes

Der POSIX-Verwalter gibt neben dem Produktnamen die Produktversion im Format Vmm.n (m,n: Ziffern) und zusätzlich den Korrekturstand im Format aso (a: Buchstabe; s,o: Ziffern) in der von IMON geforderten Schreibweise an.

3. Installation mit frei gewählter Version und mit dem von IMON voreingestelltem Korrekturstand

Der POSIX-Verwalter gibt neben dem Produktnamen die Produktversion im Format Vmm.n (m,n: Ziffern) an. Es wird entweder der höchste Korrekturstand der angegebenen Version oder - falls vor dem Start des POSIX-Installationsprogramms ein Kommando /SELECT-PRODUCT-VERSION eingegeben wurde - der durch dieses Kommando ausgewählte Korrekturstand auf POSIX installiert.

## Installation von einer privaten Ablagekennung

Der POSIX-Verwalter gibt dabei neben dem Produktnamen nur die Version im bisherigen Format mmn und den Namen der Ablagekennung an. Anschließend wird das Produkt aus der Bibliothek \$<userid>.SINLIB.<product>.mmn installiert.

## Private Programmpakete installieren

Private Programmpakete oder Programmpakete anderer Hersteller müssen, um mit dem POSIX-Installationsprogramm installiert werden zu können, in die nachfolgend beschriebene Form gebracht werden.

Das POSIX-Installationsprogramm erwartet die Produktbestandteile in einer PLAM-Bibliothek. Die ausführbaren Programme müssen als L-Elemente abgelegt sein.

Header-Files, Shell-Scripts und sonstige Bausteine wie Textdateien müssen als S-Elemente abgelegt sein.

Die PLAM-Bibliothek hat den produktspezifischen Namen SINLIB.<product>.<version>.

Außerdem werden folgende produktspezifischen Dateien als S-Element ebenfalls in der PLAM-Bibliothek benötigt:

INSTALL.<product>.<version> und DELETE.<product>.<version>

Diese Dateien beschreiben den Ablageort jedes Produktbestandteils im POSIX-Dateisystem und liefern weitere Informationen, die die Ablage betreffen in der Form:

:Element:Kennbuchstabe:Pfadname:Linkname:Zugriff:Benutzernummer:Gruppennummer

Die Spaltenbreite ist variabel. Das Trennzeichen „:“ muß auch angegeben werden, wenn kein Wert angegeben wird. Kommentarzeilen beginnen mit „#“.

Der Kennbuchstabe kennzeichnet die Installations-Teilfunktion:

- d Das Dateiverzeichnis, das im Pfadnamen angegeben ist, wird neu eingerichtet
- f Das Kommando wird unter dem angegebenen Pfadnamen angelegt
- l Für den angegebenen Linknamen wird ein Hard-Link angelegt
- m Das Kommando wird unter dem angegebenen Pfadnamen angelegt
- o Eintrag für Dateien, die entfernt werden sollen
- p Eine Prozedur (Element) wird unter dem angegebenen Pfadnamen angelegt
- r Script (Prozedur) mit dem angegebenen Pfadnamen wird ausgeführt
- s Für den angegebenen Linknamen wird ein symbolischer Link angelegt
- u Codierte T-Dateien für iconv
- v Eintrag für Dateiverzeichnisse, die entfernt werden sollen

Der Zugriff stellt die Zugriffsberechtigung für Eigentümer, Gruppe und Andere dar (oktal).

Folgende Regeln gelten für die Reihenfolge der Einträge:

- Wenn Dateiverzeichnisse angelegt werden sollen, müssen die entsprechenden Einträge am Anfang der Tabelle stehen.
- Für jedes ausführbare Programm ist nur ein einziger f-Eintrag erlaubt; er muß der erste Eintrag sein, der dieses ausführbare Programm betrifft.
- o- und v-Einträge müssen am Ende der Tabelle stehen. Zuerst müssen die Einträge für Dateien stehen, daran anschließend die Einträge für Dateiverzeichnisse.

## Protokollierung der Installation

- Logging-Datei der Paket-Installation

Die Paket-Installation wird in der Datei `/var/sadm/pkg/instlog` protokolliert.

Pro Vorgang wird ein Eintrag geschrieben, der folgende Informationen enthält:

- Kennzeichen, ob Installation oder Löschung  
(in der 1. Spalte: I (install) oder D (delete))
- Name des Produkts oder Pakets
- Version des Produktes
- Datum und Uhrzeit der Installation bzw. Löschung
- nur bei Installation: Installationsbibliothek

- Tool-Logging-Datei

Der `init`-Prozess, die POSIX-Shell und weitere Tools schreiben Meldungen in die Datei `/var/adm/messages`.

### 5.3.2 Erstinstallation

Im POSIX-Installationsprogramm wird überprüft, ob POSIX gestartet ist. Nur wenn dies nicht der Fall ist, kann die Erstinstallation des POSIX-Subsystems und seiner Dateisysteme eingeleitet werden.

Die Erstinstallation muß in folgenden Schritten unter der Benutzerkennung TSOS durchgeführt werden:

1. Installieren des Subsystems POSIX auf dem generischen root-Dateisystem

POSIX-Installationsprogramm aufrufen:

```
/CALL-PROC NAME=*LIB(LIB=$TSOS.SINPRC.POSIX-BC.<version>,ELEM=POSINST)
```

2. Anlegen und Einhängen eines neuen root-Dateisystems

Im POSIX-Installationsprogramm die Option „Install POSIX subsystem“ aufrufen und in der Maske die Angaben zur Behälterdatei und zum root-Dateisystem eintragen (siehe Seite 89).

Das root-Dateisystem muß mindestens 4096 PAM-Seiten groß sein und muß unter SYSROOT angelegt werden. Wenn mehrere Produkte installiert werden, muß es größer angelegt werden (empfohlene Größe: 20 000 PAM-Seiten). Nähere Angaben finden Sie in den entsprechenden Produkthandbüchern.

3. Anlegen und Einhängen eines var-Dateisystems (für die Verwaltungsdateien)

Im POSIX-Installationsprogramm die Option „Install POSIX subsystem“ aufrufen und in der Maske die Angaben zur Behälterdatei und zum var-Dateisystem eintragen (siehe Seite 89).

Das var-Dateisystem muß mindestens 4096 PAM-Seiten groß sein und sollte unter SYSROOT angelegt werden.

Anschließend werden wichtige Dateiverzeichnisse und Dateien automatisch aus dem generischen root-Dateisystem in das neu angelegte root- und var-Dateisystem kopiert. Die generierten Dateiverzeichnisse, Geräte- und Verwaltungsdateien werden protokolliert. Nachdem alle Dateiverzeichnisse und Dateien kopiert worden sind, ist das Arbeiten mit POSIX möglich. Das BS2000-Kommando /START-POSIX-SHELL (siehe Seite 157) kann nun eingegeben werden.

### Installation von WebServer auf BS2000/OSD

Die zu WebServer gehörenden Dateien werden während der Erstinstallation von POSIX in dem Verzeichnis */opt/httpd* abgelegt.

Ausführliche Informationen finden Sie in der Readme-Datei */opt/httpd/README*.

### 5.3.3 Upgrade-Installation

Die POSIX-Upgrade-Installation ist erforderlich, falls Sie POSIX bereits installiert haben und vorgenommene Änderungen beibehalten wollen.

Die folgenden Verfahrensschritte beschreiben sowohl den Upgrade auf einen neuen Korrekturstand von POSIX als auch einen POSIX-Versionswechsel, der z.B. durch einen Wechsel der BS2000/OSD-Version notwendig werden kann.

#### 1. Hochrüsten eines bestehenden HOME-Pubsets



##### Nur bei POSIX-Versionswechsel:

Die Upgrade-Installation braucht unbedingt gleichzeitigen Zugriff auf die POSIX-Produkt-Dateien von <oldversion> und <newversion> (Dateinamen S\*.POSIX-BC.<version>).

- Spielen Sie die SOLIS-Auslieferung der neuen POSIX-Version ein.

Bei POSIX-Versionswechsel:

- die POSIX-Produktdateien der alten POSIX-Version müssen auf dem HOME-Pubset vorhanden sein.
- Der Name des root-Dateisystem ist in der POSIX-Parameterdatei SYSSSI.POSIX-BC.<oldversion> hinterlegt. Deshalb muß die alte Parameterdatei nach dem Einspielen SOLIS-Auslieferung auf die neue Parameterdatei SYSSSI.POSIX-BC.<newversion> kopiert werden.

- Starten Sie das Subsystem POSIX mit dem Kommando  
/START-SUBSYSTEM POSIX

Die neue POSIX-Version wird dabei mit den alten Dateisystemen hochgefahren.

- Nach 'POSIX ready':

Starten Sie das POSIX-Installationsprogramm (siehe Seite 85) und wählen Sie die Option „Install packages on POSIX“.

Installieren Sie das Produkt 'POSIX-BC' mit der neuen Versionsnummer <newversion>. Beenden Sie danach das POSIX-Installationsprogramm.

- Stoppen Sie das Subsystem POSIX mit dem Kommando  
/STOP-SUBSYSTEM POSIX

- Starten Sie erneut das Subsystem POSIX (/START-SUBSYSTEM POSIX).

Damit steht die neue POSIX-Installation mit allen Kundenanpassungen zur Verfügung.

Falls ein POSIX-Versionswechsel erfolgt ist, können nun die Dateien der alten POSIX-Version gelöscht werden.

2. Erstellung eines neuen HOME-Pubsets mit weiterer Nutzung der POSIX-Daten; bzw. Wechsel der BS2000/OSD-Version
  - a) Kopieren Sie die alten root- und var-Dateisysteme sowie die alten POSIX-Produkt-Dateien S\*.POSIX-BC.<oldversion> auf das neue HOME-Pubset.
  - b) Nach dem Hochfahren des neuen HOME-Pubsets verfahren Sie weiter wie ab 1a beschrieben.

Wenn auf POSIX aufsetzende Produkte wie z.B. NFS oder POSIX-SH im Korrekturpaket ausgeliefert wurden, so ist für diese Produkte nach der POSIX-Upgrade-Installation eine POSIX-Paketinstalltion erforderlich.

### 5.3.4 POSIX-Installationsprogramm im Dialog

Das POSIX-Installationsprogramm wird aufgerufen mit:

```
/CALL-PROCEDURE NAME=*LIB(LIB=$TS0S.SINPRC.POSIX-BC.<version>,ELEM=POSINST)
```

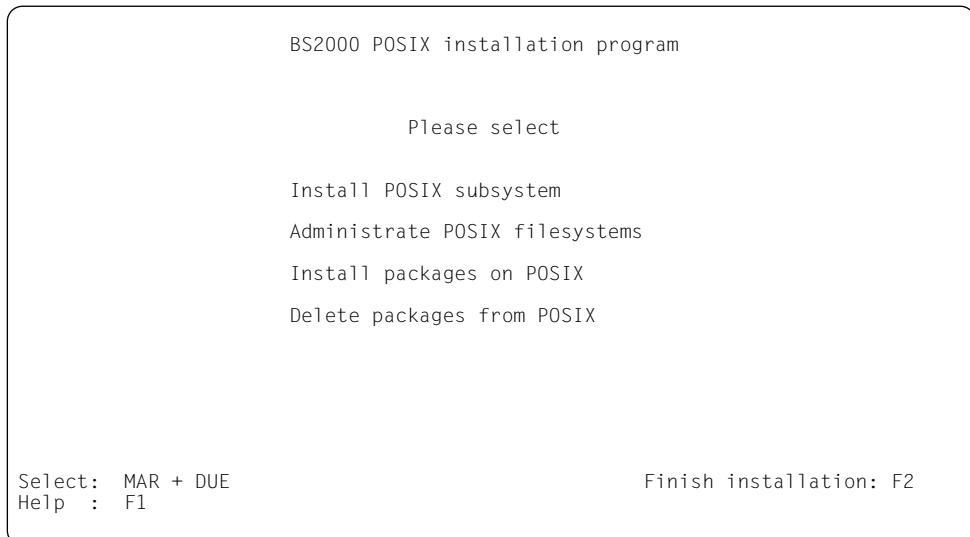


Bild 24: Hauptmaske des POSIX-Installationsprogramms

Das POSIX-Installationsprogramm stellt folgende Optionen zur Verfügung:

- Install POSIX subsystem (Subsystem POSIX neu einrichten  $\hat{=}$  Erstinstallation) (wenn das Subsystem POSIX nicht aktiv ist)
- Administrate POSIX filesystems (POSIX-Dateisystem verwalten) (wenn das Subsystem POSIX aktiv ist)
- Install packages on POSIX (Programmpakete hinzufügen) (wenn das Subsystem POSIX aktiv ist)
- Delete packages from POSIX (Programmpakete entfernen) (wenn das Subsystem POSIX aktiv ist)

Um eine der Optionen auszuwählen, müssen Sie die Option mit der Schreibmarke anwählen, durch Angabe eines Zeichens oder mit der **[MAR]**-Taste markieren und die Auswahl mit **[DUE]** bestätigen.

Mit der Funktionstaste **[F1]** erhalten Sie Hilfe zur Maske, mit der Funktionstaste **[F2]** beenden Sie das Installationsprogramm.

Meldungen und Informationen des Installationsprogramms erhalten Sie in der letzten Bildschirmzeile.

## Install POSIX subsystem (Subsystem POSIX neu einrichten)

Mit dieser Option kann ein neues Subsystem POSIX eingerichtet werden.

Das root- und das var-Dateisystem werden mit den Benutzerangaben angelegt, Dateiverzeichnisse und Dateien werden im POSIX-Dateisystem installiert und das Subsystem POSIX wird mit dem soeben angelegten root-Dateisystem gestartet.

```
Definition of BS2000 Container File
BS2000 filename: $SYSROOT.
BS2000 filesize:          PAM-Pages
POSIX filesystem? (y/n): Y
=====
Definition of POSIX filesystem
Size of filesystem:          PAM pages
POSIX mountpoint: /
Automount? (y/n): Y          Mountoptions:
Overwrite existing filesystem? (y/n):      POSIX filesystem marker (y/n): Y
=====
Save definitions: DUE
Help          : F1
                           terminate: F2
```

Bild 25: Folgemaske zu „Install POSIX subsystem“

### Angaben zur BS2000-Behälterdatei:

#### BS2000 filename

Name der PAM-Datei, die als Behälterdatei für das root- bzw. var-Dateisystem benutzt werden soll.

Der Dateiname muß die Benutzerkennung SYSROOT enthalten. Wenn die Datei noch nicht existiert, wird sie in der angegebenen Größe erzeugt.

#### BS2000 filesize

Größe der Behälterdatei in PAM-Seiten (Einheit: 2 KB).

Die Mindestgröße muß 4096 PAM-Seiten betragen.

Für neu einzurichtende Behälterdateien müssen Sie die gewünschte Größe eintragen. Wenn die Behälterdatei bereits existiert, wird die tatsächliche Größe in das Feld übernommen. In diesem Fall können Sie den Wert nicht ändern.

**POSIX filesystem? (y/n)**

Beantworten Sie die Frage mit y (ja) oder n (nein). Normalerweise soll die Behälterdatei ein POSIX-Dateisystem enthalten. In besonderen Fällen können Sie aber den Zugriff über das POSIX-Dateisystem umgehen und direkt auf den Datei-Inhalt zugreifen (*raw*-Zugriff). Die Felder für das root- und var-Dateisystem sind mit y (ja) vorbelegt.

**Angaben zum POSIX-Dateisystem:****Size of filesystem**

Dieses Feld zeigt die Größe eines bereits vorhandenen Dateisystems in der BS2000-Behälterdatei in PAM-PAGES (2 KB) an. Wenn noch kein Dateisystem existiert, wird die Größe der BS2000-Behälterdatei angezeigt.

Das Feld kann nicht überschrieben werden, da ein Dateisystem in einer BS2000-Behälterdatei immer die Größe der BS2000-Behälterdatei hat.

**POSIX mountpoint**

Dateiverzeichnis, in das das POSIX-Dateisystem eingehängt werden soll.

Sie müssen den absoluten Pfadnamen des Dateiverzeichnisses eingeben, wobei der Name mit einem Schrägstrich (/) beginnen muß. Wenn das Dateiverzeichnis noch nicht existiert, wird es vom Programm angelegt.

Die Felder für das root- und var-Dateisystem sind mit / vorbelegt.

**Automount? (y/n)**

Wenn das Dateisystem sofort und bei jedem Subsystem-Start automatisch eingehängt werden soll, müssen Sie y (ja) eintragen. Wenn Sie das Dateisystem nur einrichten, aber noch nicht benutzen wollen, müssen Sie n (nein) eintragen.

Die Felder für das root- und var-Dateisystem sind mit y (ja) vorbelegt.

**Mountoptions**

Sie können das Einhängen des Dateisystems parameterisieren. Die entsprechenden Optionen finden Sie beim Kommando *mount* im Handbuch „POSIX-Kommandos“ [2].

**Overwrite existing filesystem? (y/n)**

Beantworten Sie die Frage mit y (ja) oder n (nein).

Das Feld ist nur aktiviert, wenn die Behälterdatei bereits ein POSIX-Dateisystem enthält. Sie müssen entscheiden, ob das Dateisystem unverändert übernommen werden soll oder ob ein neues Dateisystem erzeugt werden soll.

**POSIX filesystem marker (y/n)**

Bedeutung: Dateisystem in POSIX/BS2000 erzeugt.

Die Felder für das root- und var-Dateisystem sind mit Y (ja) vorbelegt.

Wenn der Marker nicht gesetzt ist, wird das Dateisystem unter POSIX als ASCII-Dateisystem betrachtet. D.h., es findet in Abhängigkeit von der Umgebungsvariablen *IO\_CONVERSION* eine ASCII-EBCDIC-Konvertierung statt (siehe „Kopieren und Konvertieren von Dateien“ auf Seite 27).

## Administrare POSIX filesystems (POSIX-Dateisysteme verwalten)

Mit dieser Option können bereits vorhandene POSIX-Dateisysteme verwaltet, neue angelegt und bestehende gelöscht werden.

```

BS2000 POSIX filesystem table

BS2000 filename          size   filesystem
. $SYSROOT.FS.VAR         4098   Y
. $SYSROOT.FS.HOME1        4098   Y
x $SYSROOT.FS.HOME2       10002   Y

edit commands:  'a'=append  'm'=modify  'd'=delete
function keys:  F1=help   F2=terminate
command  ===> m           scroll commands: '+'/ '-'
                           more:

```

Bild 26: Folgemaske zu „Administrare POSIX filesystems“

Die Folgemaske zeigt sämtliche BS2000-Dateien, die im Subsystem POSIX als lokale Behälterdateien eingetragen sind. Mit „+“ und „-“ können Sie in der Maske blättern. Jeder Eintrag enthält:

- den BS2000-Dateinamen der Behälterdatei.
- die Größe der Behälterdatei in PAM-Seiten (Einheit: 2 KB).
- einen Vermerk, ob die Behälterdatei ein POSIX-Dateisystem enthält oder nicht (Y/N).

Im unteren Teil der Maske befindet sich ein Auswahlfeld („command“), in dem Sie eines der folgenden Verwaltungskommandos eintragen können. Nötigenfalls können in Folgemasken ergänzende Angaben gemacht werden.

### ‘a’=append

Neuen Eintrag erzeugen

Behälterdatei und ggf. Dateisystem in Liste aufnehmen. Markieren Sie zuvor den Eintrag, hinter den Sie den neuen Eintrag anfügen wollen; andernfalls wird der neue Eintrag an das Ende der Liste angehängt.

Beim Anlegen eines Dateisystems mit der Funktion append hat der Systemverwalter die Wahl, ob er das Dateisystem als „von POSIX erzeugt“ (also als EBCDIC-Dateisystem) markieren will oder nicht (ASCII-Dateisystem). Das root- und das var-Dateisystem werden bei der Erstinstallation automatisch als „von POSIX erzeugt“ markiert.

**‘m’=modify**

Markierten Eintrag ändern

Bestimmte Felder des Dateisystem-Eintrags können Sie nachträglich ändern.

**‘d’=delete**

Markierten Eintrag löschen

Es wird nur der Eintrag aus der Liste entfernt. Die Behälterdatei und das Dateisystem bleiben unverändert und können später wieder eingetragen oder gelöscht werden.

## Install packages on POSIX (POSIX-Programmpakete hinzufügen)

Mit dieser Option können POSIX-Anwenderprogramme und Programmpakete im POSIX-Dateisystem installiert werden.

Siehe „Liefer- und Installationsverfahren für POSIX-Programmpakete“ auf Seite 81.

```
BS2000 POSIX package installation

IMON support? (y/n): Y

name of product: CRTE

version of product (format Vmm.n or mmn): V02.0

correction state (format aso): C40

installation userid (no IMON support):

install: DUE
help   : F1                                         terminate: F2
```

Bild 27: Folgemaske zu „Install packages on POSIX“

**IMON support? (y/n)**

Legt fest, ob aus dem SCI installiert wird (IMON support: y) oder von einer privaten Ablagekennung (IMON support: n).

Voreinstellung ist IMON support: y.

**name of product**

Produktnname (= Name der Liefereinheit).

**version of product (format Vmm.n or mmn)**

Produktversion:

- bei 'IMON support: y' im Format Vmm.n (m,n: Ziffern) oder leer
- bei 'IMON support: n' im Format mmn (m,n: Ziffern)

**correction state (format aso)**

Nur bei 'IMON support: y' und nur zusammen mit 'version of product':

Angabe des Korrekturstandes im Format aso (a: Buchstabe; s,o: Ziffern)

Das Feld muß leer bleiben, wenn 'version of product' leer ist (siehe Fall a) im Abschnitt „Installation aus dem Software Configuration Inventory (SCI)“ auf Seite 82.

**installation userid (no IMON support)**

Nur bei 'IMON support: n' (sonst leer):

User-ID der privaten Ablagekennung

Bei fehlerhafter Eingabe (z.B. Eingabe von Zeichen in ein „leeres“ Feld) wird eine Fehlermeldung ausgegeben und die Maske wird erneut zur Änderung ausgegeben.

**i** Vor dem Installieren einer neuen Version sollte die alte Version des Programm-pakets mit der Option „Delete packages from POSIX“ gelöscht werden.

## Delete packages from POSIX (POSIX-Programmpakete entfernen)

Mit dieser Option können POSIX-Anwenderprogramme und Programmpakete entfernt werden.

Siehe „Liefer- und Installationsverfahren für POSIX-Programmpakete“ auf Seite 81.

```
BS2000 POSIX package delete

        DELETE PRODUCT
name of product: POSIX-SOCKETS

version of product: 011

delete : DUE
help   : F1
terminate: F2
```

Bild 28: Folgemaske zu „Delete packages from POSIX“

### **name of product**

Name des Produkts.

### **version of product**

Version des Produkts.



Die Basis-Shell kann nicht gelöscht werden.

Wenn beim Löschen eines Programmpakets eine Datei oder Dateiverzeichnis nicht gelöscht werden kann, wird eine Meldung ausgegeben und der Löschevorgang fortgesetzt.

### 5.3.5 Automatisierter Ablauf des POSIX-Installationsprogramms

Das POSIX-Installationsprogramm wird für den automatisierten Ablauf aufgerufen mit:

```
/CALL-PROCEDURE NAME=(LIB=$TSOS.SINPRC.POSIX-BC.<version>,ELEMENT=POSINST), -  
/ PROCEDURE-PARAMETERS=(MODE=BATCH,FILE=<parameterdatei>)
```

Beim Operanden FILE müssen Sie den Namen einer Parameterdatei angeben, die die Installationsinformationen in der nachfolgend beschriebenen Form enthält.

Eine Parameterdatei besteht aus einer Identifikationszeile, aus einer oder mehreren Anweisungszeilen und wahlweise aus Kommentarzeilen.

#### Kommentare

Kommentare und Kommentarzeilen sind optional. Sie müssen immer mit „#“ beginnen.

#### Identifikationszeile

Die erste Zeile in der Parameterdatei, die keine Kommentarzeile ist, muß die Identifikationszeile sein. Dadurch wird ein Zweig für die Installation ausgewählt:

<b>[FirstInstallation]</b>	Subsystem POSIX neu einrichten
oder	
<b>[FileSystemAdministration]</b>	POSIX-Dateisystem verwalten
oder	
<b>[PackageInstallation]</b>	POSIX-Programmpakete hinzufügen
oder	
<b>[DeletePackage]</b>	POSIX-Programmpakete entfernen

Die eckigen Klammern müssen Sie immer angeben. Die Zeichenfolge zwischen den Klammern können Sie abkürzen; die Eindeutigkeit muß aber gewährleistet sein. Groß- und Kleinbuchstaben dürfen Sie beliebig verwenden.

#### Anweisungszeilen

Nach der Identifikationszeile folgen eine oder mehrere Anweisungszeilen, die die notwendigen Parameter abhängig vom Zweig enthalten. Das Trennzeichen „;“ (Semikolon) müssen Sie angeben, auch wenn Sie für einen Parameter keinen Wert angeben.

## Install POSIX subsystem (Subsystem POSIX neu einrichten)

Identifikationszeile: **[FirstInstallation]**

Anweisungszeile: **<file>;<size>**

Dabei bedeuten:

<file> BS2000-Dateiname der Behälterdatei

<size> BS2000-Dateigröße der Behälterdatei (= Größe des Dateisystems)

Es sind genau zwei Anweisungszeilen vorgeschrieben: Die erste Anweisungszeile beschreibt das root-Dateisystem, die zweite das var-Dateisystem. Wenn mehr als zwei Anweisungszeilen angegeben sind, werden die überzähligen ignoriert. Wenn Anweisungszeilen fehlen, wird die Installation abgebrochen.

Die beiden Anweisungszeilen enthalten die Angaben zur BS2000-Behälterdatei eines POSIX-Dateisystems. Dabei sind zusätzliche Angaben, die zur vollständigen Beschreibung eines Dateisystems nötig sind, entsprechend folgender Tabelle vorgegeben:

Parameter	root-Dateisystem	var-Dateisystem
Dateisystem-Inside	Y	Y
POSIX-Dateisystem-Marker	Y	Y
POSIX-Mountpoint	/	/var
Automount	Y	Y
Mount-Optionen	-	-
Overwrite	N	N

### Beispiel

```

#
# Batch Installationsdatei
#
[FirstInstallation]           # POSIX Erstinstallation
# <file>;<size>

# Einrichten des root-Dateisystems
$SYSROOT.FS.ROOT;25000

# Einrichten des var-Dateisystems
$SYSROOT.FS.VAR;5000

```

## Administrare POSIX filesystems (POSIX-Dateisysteme verwalten)

Identifikationszeile: **[FileSystemAdministration]**

Anweisungszeile:

**<op>;<file>;<size>;<flag>;<marker>;<mntdir>;<automnt>;<mntopt>;<ov>**

Dabei bedeuten:

<b>&lt;op&gt;</b>	Editierkommando: a(ppend), m(odify) oder d(elete)
<b>&lt;file&gt;</b>	BS2000-Dateiname der Behälterdatei
<b>&lt;size&gt;</b>	BS2000-Dateigröße der Behälterdatei (=Größe des POSIX-Dateisystems)
<b>&lt;flag&gt;</b>	POSIX-Dateisystem-Inside? (Y/N)
<b>&lt;marker&gt;</b>	POSIX-Dateisystem-Marker? (Y/N)
<b>&lt;mntdir&gt;</b>	POSIX-Mountpoint
<b>&lt;automnt&gt;</b>	Automount? (Y/N)
<b>&lt;mntopt&gt;</b>	Mount-Optionen
<b>&lt;ov&gt;</b>	POSIX-Dateisystem überschreiben? (Y/N)

Jede Anweisungszeile enthält das Editierkommando und die Angaben zur BS2000-Behälterdatei und zum POSIX-Dateisystem. Sie müssen nicht alle Parameter bei jedem Editierkommando angeben. So kann z. B. beim Editierkommando m(odify) u.a. die BS2000-Dateigröße nicht geändert werden. Die folgende Tabelle zeigt, welcher Parameter bei welchem Editierkommando angegeben werden muß:

Parameter	a(ppend)	m(odify)	d(elete)
BS2000-Dateiname	x	x	x
BS2000-Dateigröße	x	-	-
POSIX-Dateisystem-Inside	dy	-	-
POSIX-Dateisystem-Marker	xy	-	-
POSIX-Mountpoint	xm	o	-
Automount	dy	o	-
Mount-Optionen	dl	o	-
POSIX-Dateisystem überschreiben	xo	-	-

Dabei bedeuten:

- wird ignoriert oder ist wirkungslos, wenn syntaktisch korrekt angegeben
- dy Standardwert ist Y, wenn der Wert fehlt
- d1 Standardwert ist die leere Zeichenkette, wenn der Wert fehlt
- o optional; wenn der Wert fehlt, gilt die aktuelle Einstellung
- x Pflichtangabe
- xy Pflichtangabe, wenn bei POSIX-Dateisystem-Inside Y angegeben ist; sonst ignoriert
- xm Pflichtangabe, wenn bei Automount Y angegeben ist; sonst ignoriert
- xo Pflichteingabe bei Overwrite-Situation; sonst ignoriert

### *Beispiel*

```
#  
# Batch Installationsdatei  
#  
[FileSystemAdministration] # POSIX-Dateisysteme verwalten  
# <op>;<file>;<size>;<flag>;<marker>;<mntdir>;<automnt>;<mntopt>;<ov>  
  
# POSIX-Dateisystem loeschen  
delete;$SYSROOT.FS.USR;  
  
# Neues Dateisystem erzeugen  
# Ein existierendes Dateisystem wird ueberschrieben  
append;$SYSROOT.FS.USR;50000;;y;/home/usr;y;;y  
  
# Existierendes Dateisystem an /home/usr einhaengen  
modify;$SYSROOT.FS.USR;;;;;/home/usr;y;;y
```

## Install Packages on POSIX (Programmpakete hinzufügen)

Identifikationszeile: **[PackageInstallation]**

Anweisungszeile: **<product>;<IMON support>;<version>;<correction state>;<userid>**

Dabei bedeuten:

**<product>** Produktname des Softwarepakets

**<IMON support>**

Legt fest, ob aus dem SCI installiert wird (Y)  
oder nicht (N)

**<version>** Produktversion des Softwarepakets

im Format Vmm.n bei Installation aus dem SCI oder

im Format mmn bei Installation von einer privaten Ablagekennung  
(m,n: Ziffern)

**<correction state>**

Korrekturstand im Format aso

(a: Buchstabe; s,o: Ziffern;

nur bei Installation aus dem SCI und nur zusammen mit <version>)

**<userid>**

User-ID der Ablagekennung

(nur bei Installation von privater Ablagekennung)

### Beispiel

```
#  
# Batch Installationsdatei  
#  
#[PackageInstallation]           # Programmpakete installieren  
# <product>;<IMON support>;<version>;<correction state>  
  
# Installation von CRTE  
CRTE;Y;V02.0;C40
```

## Delete Packages from POSIX (Programmpakete entfernen)

Identifikationszeile: **[DeletePackage]**

Anweisungszeile: **<product>;<version>**

Dabei bedeuten:

**<product>** Produktnname des Softwarepakets

**<version>** Produktversion des Softwarepakets

im Format Vmm.n oder im Format mmn (m,n: Ziffern)

### *Beispiel*

```
#  
# Batch Installationsdatei  
#  
[DeletePackage]           # Programmpakete löschen  
# <product>;<version>  
  
# Löschen von CRTE  
CRTE;V02.0
```

## 5.4 POSIX-Informationsdatei

Nach der Installation müssen Sie eventuell die POSIX-Steuerparameter an die Gegebenheiten Ihres Systems anpassen.

POSIX verwendet dazu - analog zu SINIX V5.41 - eine Informationsdatei mit einstellbaren Steuerparametern. Diese Steuerparameter bestimmen die Größe der Systemtabellen und kontrollieren dadurch die Ressourcen, die das System und die Benutzer beanspruchen können.

Jedem Steuerparameter ist ein Standard-, ein Minimal- und ein Maximalwert zugeordnet. Die Standardwerte sind so gewählt, daß das POSIX-Subsystem in beliebiger Umgebung ablaufen kann, ohne das Gesamtsystem durch übermäßigen Ressourcenverbrauch zu belasten.

### Inhalt der POSIX-Informationsdatei SYSSSI.POSIX-BC.<version>

Bei der Auslieferung der Informationsdatei ist für den Steuerparameter ROOTFSNAME kein Wert eingetragen. Der Kunde muß bei der Erstinstallation den Namen seines root-Dateisystems selbst eintragen. Die Systemkennung SYSROOT ist als Eigentümer des root-Dateisystems obligatorisch und muß deshalb nicht angegeben werden.

Numerische Parameterwerte dürfen in den Einheiten Kilo (1024) und Mega (1048567) angegeben werden.

Parameter	: Beschreibung	: Standard	Minimal	Maximal
SEGMAPSZ	: # of buffer cache entries	: 256	256	22500
HDSTNI	: # of hard disk server tasks	: 4	1	16
SHMMAX	: max. size of a shared memory segment	: 16M	131072	16M
SHMMIN	: min. size of a shared memory segment	: 1	1	1
SHMMNI	: # of shared memory headers	: 100	100	100
SHMSEG	: max. # of segments per process	: 16	6	16
SEMMAP	: # of entries in semaphore map	: 150	10	150
SEMMNI	: # of semaphore identifiers	: 150	10	150
SEMMNS	: # of semaphores in system	: 200	60	200
SEMMNU	: # of undo structures in system	: 200	30	200
SEMMSL	: max. # of semaphores per id	: 25	25	25
SEMOPM	: max. # of operations per semop call	: 20	10	20
SEMUME	: max. # of undo entries per process	: 20	10	20
SEMVMX	: semaphore maximum value	: 32767	32767	32767
SEMAEM	: adjust on exit max value	: 16384	16384	16384
MSGMAP	: # of entries in msg map	: 200	10	200

Parameter	: Beschreibung	: Standard	Minimal	Maximal
MSGMAX	: max. message size	: 2048	512	2048
MSGMNB	: max. # bytes on queue	: 16384	4096	16384
MSGMNI	: # of message queue identifiers	: 150	50	150
MSGSSZ	: msg segment size	: 8	8	8
MSGSQL	: # of system message headers	: 160	40	160
MSGSEG	: # of msg segments (MUST BE < 32768)	: 2048	1024	32768
NHBUF	: buffer cache size for metadata	: 256	32	1024
PGOVERFLOW	: overflow buffers for pageout	: 32	32	32
NPROC	: max. # of processes	: 200	50	500
NBUF	: # of I/O buffers	: 200	100	2000
MAXUP	: max. # of processes per user	: 50	15	200
BUFHWM	: high-water-mark of buffer cache	: 2000	5	2000
HEAPSZ	: size of heap-segment	: 4M	2M	4M
FDFLUSHR	: fsflush time interval	: 5	1	5
NAUTOUP	: age of a delayed-write buffer	: 60	10	120
FLCKREC	: max. # of active file records locks	: 1000	100	2000
UFSNINODE	: # of inodes	: 1000	600	1000
NRNODE	: max. # of incore remote nodes (nfs)	: 600	400	600
ROOTFSNAME	: name of root-file system	:		
NOFILES	: max. # of file descriptors	: 2048	20	2048
FILESIZE	: max. size of file	: UNLIMITED	64	UNLIMI.
HDPTNI	: # of partition table entries	: 256	16	256
NPBUF	: number of physical I/O buffers	: 20	20	40
MINPAGEFREE	: pageout daemon / min. # of free pages	: 0	0	0
KMAHWM	: kma daemon: kmem high water mark	: 2M	1M	2M
NOTTY	: max. # of ttys	: 16	4	4096
NOPTY	: max. # of ptys	: 16	256	256
NOSTTY	: max. # of sttys	: 16	4	4
BINDANY	: BCAM / mode flag	: 0	0	1
PORTMON	: port monitoring (nfs)	: 1	0	1
DBLPOOL	: memory pool in class 6 memory	: 0	0	1024
DBLSTATE	: state of the loader	: 0	0	1
MAXTIMERC	: max. wait time for rc term procs	: 660	120	1200

## Standardwerte der Steuerparameter ändern

In vielen Fällen sind die Standardwerte der Steuerparameter ausreichend. Manchmal kann es aber sinnvoll sein, daß der BS2000-Systemverwalter Steuerparameter an die spezielle POSIX-Anwendung und an den Ressourcenvorrat des Gesamtsystems anpaßt. Im folgenden Teil sind die Steuerparameter aufgeführt, für die eine Änderung sinnvoll sein kann. Zusätzlich ist bei jedem Steuerparameter die Bedeutung angegeben.

### Allgemeine Systemparameter

#### **NPROC**

Maximale Anzahl der Benutzerprozesse, die im System erlaubt sind.

#### **MAXUP**

Maximale Anzahl der Prozesse, die ein nichtprivilegierter Benutzer gleichzeitig starten kann (nicht pro Terminal, sondern insgesamt).

#### **FLCKREC**

Anzahl der vom System verwendeten Sperrstrukturen für Datensätze (recordlocks).

### Dateisystemparameter

#### **FDFLUSHR**

Zeitintervall (in Sekunden) zwischen zwei Aktivierungen eines Prozesses.

*fsflush* schreibt Daten aus dem Cache-Puffer auf die Festplatte und stellt damit die Konsistenz der Daten auf der Festplatte sicher. Ein kleiner Wert für FDFLUSHR bringt größere Sicherheit gegen Datenverlust bei einem Systemausfall, geht aber zu Lasten der Systemleistung.

#### **NAUTOUP**

Angabe (in Sekunden), wie lange ein Puffer im Speicher „altern“ muß, bevor er durch *fsflush* zurückgeschrieben wird. Dieser Wert betrifft nur den Inhalt des Cache-Puffers.

#### **NHBUF**

Anzahl von Hash-Ankern für den schnellen Zugriff auf Puffer des Cache-Puffers über Geräte- und Blocknummern.

#### **NBUF**

Anzahl von Ein-/Ausgabepuffern des Cache-Puffers, die vom Systemkern zugewiesen werden, wenn keine mehr frei sind.

#### **BUFHWM**

Größe des Speichers (in Kilobytes), der durch die Ein-/Ausgabepuffer belegt werden kann.

**SEGMAPSZ**

Größe des Cache-Puffers (in Einheiten von 8 KB). Die Größe des Cache-Puffers bestimmt den Ein-/Ausgabedurchsatz eines Rechners. SEGMAPSZ gibt keinen reservierten Bereich im physikalischen Speicher an, sondern lediglich ein Maximum in Einheiten von 8 KB, bis zu dem sich der Cache-Puffer ausdehnen kann. Im Normalfall werden weniger Seiten belegt, als unter SEGMAPSZ angegeben sind.

SEGMAPSZ ist eine der wichtigsten Steuerparameter. Wenn SEGMAPSZ zu klein ist, kann der Systemkern die Plattenzugriffe nicht optimal puffern. Wenn er zu groß ist, gibt es im BS2000-Betrieb Probleme, weil kein Klasse-4-Speicher mehr verfügbar ist (ggf. muß der Parameter SYSSIZE angepaßt werden, siehe Seite 78).

**UFSNINODE**

Maximale Anzahl von UFS-Indexeinträgen im Systemkern.

**NRNODE**

Maximale Anzahl von NFS-rnode-Strukturen. Dies sind spezielle Deskriptoren für offene Dateien aus NFS-Dateisystemen, d.h. diese Dateien liegen auf fernen Rechnern.

**Steuerparameter für die Interprozeßkommunikation**

Nachrichten-Warteschlangen und Semaphore werden über sogenannte Resourcemap veraltet. Resourcemap führen darüber Buch, wieviel Speicherplatz von Meldungen und Semaphoren verbraucht wurde. Die Anzahl der belegten Einträge einer Resourcemap zu einem bestimmten Zeitpunkt ist ein Maß für die aktuelle Stückelung des für Meldungen verfügbaren Speicherbereichs oder der verfügbaren Semaphore.

Wenn Steuerparameter wie MSGSEG oder SEMMNS erhöht werden, sollte die Größe der entsprechenden Resourcemap ebenfalls erhöht werden.

**MSGMAP**

Anzahl der Einträge in der Resourcemap für Nachrichten-Warteschlangen.

**MSGMAX**

Maximale Größe einer Meldung (in Bytes).

**MSGMNB**

Maximale Gesamtgröße aller Meldungen einer Nachrichten-Warteschlange (in Bytes).

**MSGMNI**

Maximale Anzahl von Nachrichten-Warteschlangen systemweit.

**MSGSSZ**

Minimale Zuweisungsgröße für Meldungsspeicher (Segmentgröße in Bytes).

**MSGTQL**

Anzahl der Nachrichtenköpfe im System. Sie entspricht der Anzahl der ausstehenden Meldungen.

**MSGSEG**

Anzahl von Meldungs-Segmenten im System. Wenn der Wert von MSGSSZ mit dem Wert von MSGSEG multipliziert wird, erhält man den gesamten Speicherplatz, der für Meldungsdaten zur Verfügung steht.

**SEMMAP**

Anzahl der Einträge in der Resourcemap für Semaphor-Sätze.

**SEMMNI**

Maximale Anzahl von Semaphor-Sätzen.

**SEMMNS**

Maximale Anzahl von Semaphoren im System.

**SEMMNU**

Maximale Anzahl von Prozessen mit noch ausstehenden Undo-Operationen. Prozesse können festlegen, ob ihre Semaphor-Aktionen bei Prozeßende automatisch rückgängig gemacht werden.

**SEMMSL**

Maximale Anzahl von Semaphoren pro Satz.

**SEMOPM**

Maximale Anzahl von Semaphor-Operationen, die pro *semop(2)*-Systemaufruf ausgeführt werden können.

**SEMUME**

Maximale Anzahl von Undo-Operationen pro Prozeß.

**SEMVMX**

Maximaler Wert für eine Semaphore.

**SEMAEM**

Maximaler Undo-Wert für eine Semaphore.

**SHMMAX**

Maximale Größe eines gemeinsam nutzbaren Speicherbereichs (in Bytes).

**SHMMIN**

Minimale Größe eines gemeinsam nutzbaren Speicherbereichs (in Bytes).

**SHMMNI**

Maximale Anzahl von gemeinsam nutzbaren Speicherbereichen.

**SHMSEG**

Maximale Anzahl von gemeinsam nutzbaren Speicherbereichen, die ein Prozeß gleichzeitig verwendet.

## Spezielle Parameter für POSIX

### **NOTTY**

Maximale Anzahl von tty's, die POSIX unterstützt.

### **NOPTY**

Maximale Anzahl physikalischer Terminals.

### **NOSTTY**

Maximale Anzahl von stty's, die POSIX unterstützt (stty ist ein Wertesatz, mit dem der TTY-Monitor einen Datensatz des Ports überschreibt, z.B. nach Erhalt eines Signals).

### **BINDANY**

Der Parameter BINDANY entschärft ein mögliches Sicherheitsproblem im Netz; er spielt nur beim Einsatz von SOCKETS eine Rolle.

SOCKETS arbeitet auf Basis von BCAM. Deshalb ist es nicht möglich, daß ein *bind()*-Aufruf, in dem eine bestimmte Schnittstelle angegeben wird, nur eine Verbindung zu dieser Schnittstelle schafft.

Der Parameter BINDANY bewirkt in sicheren Systemen (Parameterwert 0), daß jeder *bind()*-Aufruf mit errno EADDRNOTAVAIL zurückgewiesen wird. In Netzmengen, in denen dieser Sicherheitsaspekt keine Rolle spielt, kann der Parameter auf 1 gesetzt werden. Ein *bind()*-Aufruf auf eine bestimmte Schnittstelle wird dann an BCAM weitergereicht und damit implizit auf alle Schnittstellen durchgeführt.

### **PORTRMON**

Ein- oder Ausschalten der Portüberwachung für NFS (0=ausgeschaltet, 1=eingeschaltet).

### **MAXTIMERC**

Maximale Wartezeit für den Ablauf der rc-Beendigungsprozeduren bei der POSIX-Terminierung.

### **DBLPOOL**

Um die Ladevorgänge mit Hilfe des POSIX-Laders zu beschleunigen, muß hier ein Wert größer als Null eingegeben werden (in Megabyte).

### **DBLSTATE**

Zustand des Laders: 0=ausgeschaltet (Default); 1=eingeschaltet

## 6 POSIX steuern, Dateisysteme verwalten

Dieses Kapitel wendet sich an die Systemverwalter von BS2000 und POSIX. Es informiert sie über

- das Starten und Beenden von POSIX
- das Verwalten von Dateisystemen

Die folgende Tabelle zeigt die notwendigen Privilegien für die POSIX-Verwaltungsaufgaben und dafür vorgesehene Kommandos bzw. Programme.

Aufgabe	Privileg	Kommando etc.	Eingabe in
POSIX starten und beenden	SUBSYSTEM-MANAGEMENT	/START-SUBSYSTEM SUBSYSTEM-NAME=POSIX /STOP-SUBSYSTEM SUBSYSTEM-NAME=POSIX	BS2000
POSIX-Dateisysteme einrichten, ändern und löschen	TSOS mit Root-Berechtigung	POSIX-Installationsprogramm	BS2000
POSIX-Dateisysteme ein- und aushängen	Root-Berechtigung	mount, mountall; umount, umountall	POSIX-Shell

## 6.1 POSIX starten

Folgende Voraussetzungen müssen erfüllt sein, damit ein Benutzer mit dem Privileg SUBSYSTEM-MANAGEMENT POSIX starten kann:

- Das Subsystem POSIX muß installiert sein (siehe Kapitel „POSIX installieren“ auf Seite 77).
- Die Vorbereitungen für den Init-Prozeß müssen durchgeführt sein (siehe Seite 79).
- Eventuell muß die POSIX-Informationsdatei angepaßt werden (siehe Seite 101).

Der Name der Behälterdatei, in der sich das root-Dateisystem befindet, muß mit dem Steuerparameter ROOTFSNAME der POSIX-Informationsdatei übereinstimmen.

Bei einer Erstinstallation wird der Name des neu erzeugten root-Dateisystems in die POSIX-Informationsdatei eingetragen. Deshalb ist in diesem Fall keine Kontrolle auf Übereinstimmung nötig.

- Der schreibende Zugriff auf die Behälterdatei des root-Dateisystems und aller anderen Dateisysteme, die während des POSIX-Starts eingehängt werden sollen, muß möglich sein (Attribut ACCESS=\*WRITE im Dateikatalog).
- POSIX muß im Subsystemkatalog eingetragen sein.

POSIX wird entweder automatisch nach erfolgter Erstinstallation gestartet oder explizit mit dem BS2000-Kommando /START-SUBSYSTEM SUBSYSTEM-NAME=POSIX.

Wenn das Starten des Subsystems POSIX erfolgreich abgeschlossen wurde, erscheint an der Konsole folgende Meldung:

POS4100: INIT: THE POSIX SUBSYSTEM IS READY.

Wenn das Starten nicht erfolgreich abgeschlossen wurde, weil z.B. der gestartete Init-Prozeß nicht beendet werden konnte, können Sie die Ursache der Protokolldatei des Init-Prozesses \$SYSROOT.SYSLOG.POSIX-BC.<version>.INIT entnehmen.

## Unterstützung von rc-Prozeduren

POSIX unterstützt zwar nicht den Runlevel-Mechanismus des Native UNIX, jedoch können in Anlehnung an UNIX rc-Prozeduren definiert werden, die beim Starten und Beenden von POSIX automatisch ablaufen. Wie bei UNIX müssen die rc-Prozeduren, die beim Start aktiviert werden sollen, im Verzeichnis `/etc/rc2.d` abgelegt werden und die Prozeduren, die beim Beenden aktiviert werden, im Verzeichnis `/etc/rc0.d`.

Insbesondere wird der *inetd*-Dämon, der für *rlogin* benötigt wird, automatisch gestartet und beendet. Wenn BCAM explizit (mit `/BCEND`) beendet wird, während POSIX noch läuft, wird der *inetd*-Dämon beendet. Bei Bedarf muß er in der POSIX-Shell mit folgendem Aufruf neu gestartet werden: `/sbin/inetd`

Der *shmd*-Dämon, der die Shared-Memory-Funktionalität von POSIX bereitstellt, wird ebenfalls mit einem rc-Script automatisch gestartet. Während der POSIX-Session darf er nicht beendet werden. Der *shmd*-Dämon wird durch die reguläre POSIX-Terminierung beendet.

In POSIX-Systemen, auf denen NFS installiert ist, wird NFS über rc-Scripts automatisch gestartet und beendet.

Wenn die POSIX-Shell ab V1.1A installiert ist, wird der *cron*-Dämon ebenfalls über rc-Prozeduren automatisch gestartet und beendet.

Die Dämonen, die im Zusammenhang mit der Einführung der rc-Prozeduren neu eingerichtet werden, und die Hilfsdämonen zum Starten und Beenden von POSIX laufen unter dem Jobnamen „init“ ab. Dadurch ist der Jobname „init“ nicht mehr wie bisher ausschließlich dem init-Prozeß vorbehalten.

Während des Startens und Beendens von POSIX treten also zusätzliche Programmlademeldungen dieser Prozesse auf, eventuell nach der Meldung  
POS4100 INIT: POSIX READY (im Falle des Starts).

Die maximale Wartezeit für den Ablauf der rc-Beendigungsprozeduren während der POSIX-Terminierung ist über den neuen Parameter MAXTIMERC in der Informationsdatei (siehe Seite 106) steuerbar.

Beim Definieren eigener rc-Beendigungsprozeduren ist zu berücksichtigen, daß die POSIX-Terminierung maximal MAXTIMERC Sekunden auf die Durchführung der kunden-spezifischen und offiziellen rc-Prozeduren wartet. Anschließend wird die POSIX-Terminierung abnormal weitergeführt, d.h. alle rc-Prozeduren oder Anwenderprogramme, die zu diesem Zeitpunkt noch laufen, werden abgebrochen.

## 6.2 POSIX beenden

POSIX wird normalerweise durch einen Benutzer mit dem Privileg SUBSYSTEM-MANAGEMENT beendet oder automatisch beim Shutdown des BS2000-Betriebssystems. Wenn ein schwerwiegender Fehler auftritt, wird POSIX abnormal beendet.

### Subsystem POSIX durch BS2000-Systemverwalter beenden

Der BS2000-Systemverwalter beendet das Subsystem POSIX mit dem Kommando

```
/STOP-SUBSYSTEM SUBSYSTEM-NAME=POSIX
```

Dieses Kommando ist nur mit dem Privileg SUBSYSTEM-MANAGEMENT ausführbar.

Wenn zum Zeitpunkt der POSIX-Beendigung noch Benutzer mit dem Subsystem POSIX verbunden sind, bricht das DSSM die Beendigung ab. Sie können dann die Beendigung mit folgendem Aufruf erzwingen:

```
/STOP-SUBSYSTEM SUBSYSTEM-NAME=POSIX, -  
/ SUBSYSTEM-PARAMETER='FORCED-BY-SUBSYSTEM'
```

Die Zeichenkette FORCED-BY-SUBSYSTEM darf nicht abgekürzt werden.

Das Subsystem POSIX kann nicht mit dem Kommando /HOLD-SUBSYSTEM POSIX angehalten werden. Das Kommando wird zurückgewiesen; die laufende POSIX-Sitzung wird dadurch nicht beeinträchtigt.

Die Beendigung des Subsystems POSIX wird an der Konsole gemeldet:

```
POS3010: SUBSYSTEM POSIX HAS BEEN TERMINATED.
```

### Subsystem POSIX beim Shutdown des BSS2000-Betriebssystems beenden

DSSM beendet das Subsystem POSIX während des Shutdowns implizit, damit die POSIX-Dateien in einen konsistenten Zustand gebracht werden.

### Abnormale Beendigung von POSIX

Bei einem schwerwiegenden Fehler wird POSIX abnormal beendet. Dabei arbeiten die BS2000-Subsystem-Verwaltung und POSIX eng zusammen. Alle Programme, die POSIX verwenden, werden abnormal beendet. Die von POSIX belegten BS2000-Ressourcen werden freigegeben.

Wenn sich der Init-Prozeß während einer POSIX-Sitzung beendet, wird eine abnormale POSIX-Beendigung eingeleitet, da der Init-Prozeß eine zentrale Steuerungsfunktion in POSIX hat und deshalb für einen fehlerfreien Ablauf unentbehrlich ist.

## 6.3 Dateisysteme verwalten

Mehrere POSIX-Dateisysteme können zusammen einen Dateibaum bilden. Während einer POSIX-Session sind immer mindestens zwei Dateisysteme eingehängt: Das root-Dateisystem und das var-Dateisystem (siehe Abschnitt „Erstinstallation“ auf Seite 85).

Das root-Dateisystem hat die höchste Hierarchie im Dateibaum. Der BS2000-Systemverwalter muß bei der Installation angeben, welches POSIX-Dateisystem das root-Dateisystem sein soll. Das root-Dateisystem wird automatisch beim Starten des Subsystems POSIX geöffnet.

Ein Dateibaum kann durch das Verbinden zusätzlicher Dateisysteme mit dem root-Dateisystem noch erweitert werden. Dieser Vorgang wird „Einhängen“ genannt. Das Einhängen geschieht

- automatisch beim POSIX-Start, wenn das Dateisystem in der Datei `/etc/vfstab` bzw. bei „Administrate POSIX file systems“ mit dem Attribut `automount=yes` definiert ist oder
- explizit mit dem Kommando `mount`. Nur POSIX-Verwalter können POSIX-Dateisysteme einhängen.

Als Einhängepunkt kann jedes Dateiverzeichnis mit Ausnahme des root-Verzeichnisses im Dateibaum gewählt werden.

POSIX-Verwalter können eingehängte Dateisysteme mit dem Kommando `umount` wieder aushängen.



Mini-POSIX-Dateisysteme und Dateisysteme, die mit NFS V1.0 erstellt wurden, können zwar eingehängt werden, aber sie können nicht als root-Dateisystem verwendet werden.

Während einer laufenden POSIX-Session dürfen folgende Pubsets nicht exportiert werden: Das Pubset, auf dem sich das root-Dateisystem befindet, sowie alle Pubsets, auf denen sich eingehängte Dateisysteme befinden.

### 6.3.1 Lokale POSIX-Dateisysteme verwalten

Lokale POSIX-Dateisysteme können mit dem POSIX-Installationsprogramm eingerichtet, geändert und gelöscht werden. Dazu steht Benutzern mit dem Privileg TSOS und zusätzlicher Root-Berechtigung die Unterfunktion *Administre POSIX filesystems* zur Verfügung (siehe Seite 91).

Lokale POSIX-Dateisysteme, die POSIX mit dem POSIX-Installationsprogramm bekannt gemacht wurden, kann ein Root-Berechtigter ein- oder aushängen. Zum Einhängen dienen die POSIX-Kommandos *mount* und *mountall*, zum Aushängen die POSIX-Kommandos *umount* und *umountall*. Sie sind im Handbuch „POSIX-Kommandos“ [2] ausführlich beschrieben.

### 6.3.2 Verteilte Dateisysteme verwalten

Das Softwareprodukt NFS ermöglicht verteilte Dateisysteme in einem heterogenen Rechnernetz. Verteilte Dateisysteme bedeutet:

- Lokale Datenbestände können Sie für die Bearbeitung an fernen Rechnern bereitstellen. Sie können beliebige Ausschnitte aus der Hierarchie des POSIX-Dateisystems bereitstellen. Die bereitgestellten Ausschnitte dürfen sich aber nicht überlappen. Für das Bereitstellen und Zurücknehmen der Bereitstellung können Sie die Kommandos *share*, *shareall*, *unshare* und *unshareall* bzw. die Datei */etc/dfstab* verwenden.
- Von fernen Rechnern bereitgestellte Datenbestände können Sie am lokalen Rechner im POSIX-Dateisystem einhängen und bearbeiten. Der Benutzer merkt nicht, daß sich das eingehängte Dateisystem physikalisch auf einem anderen Rechner befindet. Er kann mit den Dateien dieses Dateisystems so arbeiten, als wenn sie sich im lokalen POSIX-Dateisystem befinden würden.

Für das Ein- und Aushängen von Datenbeständen, die ferne Rechner bereitstellen, können Sie die NFS-spezifischen Ausprägungen der Kommandos *mount*, *mountall*, *umount*, *umountall* bzw. die Datei */etc/vfstab* verwenden.

Näheres dazu steht im Handbuch „NFS“ [12].

### 6.3.3 Dateisystem auf Konsistenz prüfen

Mit dem POSIX-Kommando *fsck* kann die Konsistenz eines Dateisystems geprüft werden. Inkonsistenzen können im Dialog mit dem Benutzer korrigiert werden.

Das POSIX-Kommando *fsck* ist im Handbuch „POSIX-Kommandos“ [2] ausführlich beschrieben.

## 7 POSIX-Benutzer verwalten

Dieses Kapitel wendet sich an BS2000-Systemverwalter, BS2000-Gruppenverwalter und POSIX-Verwalter.

Jeder BS2000-Benutzer ist gleichzeitig auch POSIX-Benutzer. Außer einer BS2000-Benutzererkennung mit gültigen individuellen POSIX-Benutzerattributen (siehe Seite 117) sind keine weiteren Bedingungen zu erfüllen, um Zugang zu POSIX und seinen Schnittstellen zu erhalten.

Die POSIX-Benutzerverwaltung ist in die BS2000-Benutzerverwaltung integriert. Dieses Kapitel beschreibt die Schnittstellen zur Verwaltung der POSIX-Benutzerattribute einer BS2000-Benutzererkennung. Diese Schnittstellen sind Bestandteil des Bausteins SRPM (**S**ystem **R**esources and **P**rivileges **M**anagement), der im BS2000-Grundausbau und im Softwareprodukt SECOS enthalten ist. Näheres zu SRPM finden Sie in den Handbüchern „Einführung in die Systembetreuung“ und „SECOS“. Das Softwareprodukt SECOS muß nicht installiert sein, um mit POSIX arbeiten zu können.

## 7.1 Übersicht über Privilegien und Aufgaben

Für POSIX wird das Privileg POSIX-ADMINISTRATION neu eingeführt. Inhaber dieses Privilegs werden in diesem Handbuch kurz POSIX-Verwalter genannt. Sie haben folgende Aufgaben und Rechte:

- Verwaltung der POSIX-Benutzerattribute aller BS2000-Benutzerkennungen auf allen Pubsets (siehe Seite 117)
- Verwaltung der Standardwerte für die POSIX-Benutzerattribute auf allen Pubsets (siehe Seite 124)

Das Privileg POSIX-ADMINISTRATION ist automatisch an die Systemkennung SYSROOT geknüpft. Dieses Privileg kann SYSROOT nicht entzogen werden.

Der Sicherheitsbeauftragte (Privileg SECURITY-ADMINISTRATION) kann das Privileg POSIX-ADMINISTRATION auch anderen BS2000-Benutzerkennungen verleihen und entziehen. Dazu wird das Softwareprodukt SECOS benötigt.

SYSROOT ist das POSIX-Gegenstück zur Systemverwalterkennung *root* im UNIX/SINIX. SYSROOT wird beim Start des BS2000-Systems eingerichtet und erhält automatisch die Benutzernummer 0. SYSROOT kann keine andere Benutzernummer zugewiesen werden.

Inhaber des Privilegs USER-ADMINISTRATION erhalten zusätzlich die Berechtigung, die POSIX-Benutzerattribute zu verwalten. Sie sind diesbezüglich dem POSIX-Verwalter gleichgestellt.

Die Berechtigung des Gruppenverwalters der Gruppe \*UNIVERSAL wird auf die POSIX-Benutzerattribute ausgedehnt. Er ist bei der Verwaltung der POSIX-Benutzerattribute auf dem von ihm verwalteten Pubset den Inhabern des Privilegs USER-ADMINISTRATION gleichgestellt. Deshalb gelten für ihn nicht die im folgenden aufgeführten Einschränkungen für Gruppenverwalter seiner Hierarchie.

Gruppenverwalter dürfen ebenfalls die POSIX-Benutzerattribute verwalten. Allerdings gelten für sie folgende Einschränkungen:

- Sie können nicht die Standardwerte für die POSIX-Benutzerattribute verwalten.
- Die Art der POSIX-Benutzerattribute, die ihrer Verwaltung unterstellt sind, hängt von ihrer Autorisierung ab (ADM-AUTHORITY).
- Der Wertebereich der POSIX-Benutzerattribute ist für sie eingeschränkt.
- Sie können nur die Gruppen- und Untergruppenmitglieder verwalten, die ihnen unterstellt sind.

Die folgende Tabelle gibt Ihnen einen Überblick über die Aufgaben und Tätigkeiten, die im Zusammenhang mit der POSIX-Benutzerverwaltung anfallen. Dazu sind bestimmte Privilegien erforderlich. Die Aufgaben müssen entweder auf der BS2000-Ebene, auf der Shell-Ebene oder auf beiden Ebenen erledigt werden.

Aufgabe/Tätigkeit	Privileg	Kommando etc.	Eingabe in	siehe
POSIX-Status anzeigen	SUBSYSTEM-MANAGEMENT	/SHOW-POSIX-STATUS	BS2000	Seite 144
BS2000-Kennungen das Privileg POSIX-ADMINISTRATION verleihen oder entziehen	SECURITY-ADMIN.	/SET-PRIVILEGE /RESET-PRIVILEGE	BS2000	Handbuch „SECOS“ [13]
POSIX-Benutzerattribute vergeben	USER-ADMIN. oder POSIX-ADMIN. oder BS2000-Gruppenverwalter (mit Einschränkungen)	/MODIFY-POSIX-USER-ATTRIBUTES /SHOW-POSIX-USER-ATTRIBUTES	BS2000	7.2
Einer BS2000-Benutzerkennung eine individuelle Benutzernummer zuordnen	USER-ADMIN.	/MODIFY-POSIX-USER-ATTRIBUTES	BS2000	7.3
POSIX-Gruppen im BS2000 verwalten	USER-ADMIN. oder POSIX-ADMIN. oder Gruppenverwalter	/MODIFY-POSIX-USER-ATTRIBUTES: Benutzerattribut GROUP-NUMBER	BS2000	7.4, 7.5
POSIX-Gruppen in POSIX verwalten	Root-Berechtigung	Datei /etc/group	POSIX-Shell	7.4, 7.5
Neue POSIX-Benutzer eintragen (in Systemen mit SDF-P)	USER-ADMIN.	/ADD-USER und Prozedur POSADDUS	BS2000	7.5
Neue POSIX-Benutzer eintragen (in Systemen ohne SDF-P)	USER-ADMIN. und zusätzlich Root-Berechtigung	/ADD-USER <i>adduser; ...</i>	BS2000 und POSIX-Shell	7.5
Standardwerte für POSIX-Benutzerattribute festlegen	USER-ADMIN. oder POSIX-ADMIN. oder BS2000-Gruppenverwalter (mit Einschränkungen)	/MODIFY-POSIX-USER-DEFAULTS /SHOW-POSIX-USER-DEFAULTS	BS2000	7.6
Zugangsberechtigung für den Benutzer eines fernen Rechners erteilen	USER-ADMIN. oder BS2000-Gruppenverwalter (mit Einschränkungen)	/SET-LOGON-PROTECTION /MODIFY-LOGON-PROTECTION /SHOW-LOGON-PROTECTION	BS2000	7.7

Aufgabe/Tätigkeit	Privileg	Kommando etc.	Eingabe in	siehe
Abrechnungsnummer für den Systemzugang über einen fernen Rechner eintragen	USER-ADMIN. oder BS2000-Gruppenverwalter (mit Einschränkungen)	/ADD-USER /MODIFY-USER-ATTRIBUTES /SHOW-USER-ATTRIBUTES	BS2000	7.8
POSIX-Benutzer löschen	POSIX-ADMIN.	/MODIFY-POSIX-USER-ATTRIBUTES	BS2000	7.9
POSIX-Benutzer löschen	Root-Berechtigung	rmdir	POSIX-Shell	7.9
Informationen über Einträge im Benutzerkatalog für die eigenen Benutzerkennungen ausgeben	STD-PROCESSING	/SHOW-USER-ATTRIBUTES /SHOW-POSIX-USER-ATTRIBUTES	BS2000	Seite 156 Seite 145
Benutzerinformationen per Programm lesen		Makro SRMUINF		7.10

## 7.2 POSIX-Benutzerattribute vergeben

Die POSIX-Benutzerattribute charakterisieren den Benutzer, treffen Voreinstellungen und legen Berechtigungen fest. POSIX-Benutzerattribute sind: *Benutzernummer*, *Gruppennummer*, *Kommentar*, *Login-Verzeichnis* und *Programm*. Sie entsprechen den Einträgen im SINIX-Benutzerkatalog */etc/passwd*. Die Datei */etc/passwd* existiert aber in POSIX nicht.

Die POSIX-Benutzerattribute werden beim Einrichten einer BS2000-Benutzerkennung mit Standardwerten belegt (siehe Seite 124). Die POSIX-Benutzerattribute einer BS2000-Benutzerkennung können mit dem Kommando */MODIFY-POSIX-USER-ATTRIBUTES* (siehe Seite 131) geändert werden.

Bereits existierenden BS2000-Benutzerkennungen wird beim First-Start oder beim Versionsumstieg automatisch die Standard-Benutzernummer zugeordnet (siehe Seite 119). Die Standard-Benutzernummer einer existierenden BS2000-Benutzerkennung kann mit dem Kommando */MODIFY-POSIX-USER-ATTRIBUTES* (siehe Seite 131) geändert werden.

Die POSIX-Benutzerattribute sind Bestandteil des BS2000-Benutzereintrags im BS2000-Benutzerkatalog *SYSSRPM*.

### Benutzernummer

Sie legt unter POSIX fest, wer der Eigentümer von Dateien und Dateierzeichnissen ist. Im Gegensatz zum BS2000 ist hier die BS2000-Benutzerkennung - oder besser der Login-Name - zweitrangig. Deshalb muß jeder BS2000-Benutzerkennung, die POSIX nutzen möchte, eine individuelle Benutzernummer zugeordnet werden (siehe Abschnitt „Einer BS2000-Benutzerkennung eine individuelle Benutzernummer zuordnen“ auf Seite 119).

Eine Sonderrolle hat die Benutzernummer 0: Sie gibt ihrem Inhaber - zusammen mit der Gruppennummer 0 - die POSIX-Verwalter-Berechtigung, die im folgenden kurz Root-Berechtigung genannt wird. Die Systemkennung *SYSROOT* hat standardmäßig die Root-Berechtigung. Die Systemkennung *TSOS* erhält bei der Erstinstallation automatisch die Root-Berechtigung.

### Gruppennummer

Sie legt die Zugehörigkeit zu einer POSIX-Gruppe fest. Diese POSIX-Gruppe erhält die Zugriffsrechte der Benutzerklasse „Gruppe“ für alle POSIX-Dateien, die dieser Benutzer erstellt.

Der Gruppennummer kann durch einen Eintrag im POSIX-Gruppenkatalog */etc/group* ein Gruppenname zugeordnet werden (siehe Seite 121).

## Kommentar

An dieser Stelle kann ein Kommentar zum Eigentümer der BS2000-Benutzerkennung eintragen werden.

## Login-Verzeichnis

Es bestimmt den absoluten Pfadnamen des Dateiverzeichnisses, in das der Benutzer automatisch gelangt, wenn er mit POSIX verbunden wird. Dies ist im Falle

- eines gemischten Programms, das aus dem BS2000 aufgerufen wird, der erste Aufruf einer POSIX-Schnittstelle (POSIX-SVC's).
- eines Benutzers der POSIX-Shell die Verarbeitung des Kommandos /START-POSIX-SHELL
- eines *rlogin*-Aufrufs die Verarbeitung des *rlogin*.

## Programm

Dieses POSIX-Benutzerattribut bezeichnet den Namen des Programms, das gestartet wird, nachdem der Benutzer das Kommando /START-POSIX-SHELL (siehe Seite 157) aufgerufen hat.

## 7.3 Einer BS2000-Benutzerkennung eine individuelle Benutzernummer zuordnen

Eine BS2000-Benutzerkennung wird unter POSIX über die Benutzernummer identifiziert. Deshalb muß jeder BS2000-Benutzerkennung, von der aus POSIX nutzbar sein soll, eine Benutzernummer (siehe Seite 124) zugeordnet werden:

- Jeder existierenden BS2000-Benutzerkennung wird beim First-Start oder beim Versionsumstieg automatisch die Standard-Benutzernummer zugeordnet.
- Jede neu einzurichtende BS2000-Benutzerkennung erhält bereits bei ihrer Definition die Standard-Benutzernummer.

Dadurch gibt es eine Vielzahl von BS2000-Benutzerkennungen, die alle dieselbe Standard-Benutzernummer haben.

POSIX-Verwalter und BS2000-Systemverwalter können den Wert der Standard-Benutzernummer mit dem BS2000-Kommando `/MODIFY-POSIX-USER-DEFAULTS` festlegen. Sie müssen jeder BS2000-Benutzerkennung anstelle der Standard-Benutzernummer eine individuelle Benutzernummer zuordnen, bevor POSIX unter dieser BS2000-Benutzerkennung genutzt werden kann. Dazu steht das BS2000-Kommando `/MODIFY-POSIX-USER-ATTRIBUTES` zur Verfügung. Es wird eine Warnung ausgegeben, wenn eine Benutzernummer mehrfach vergeben wird, außer wenn es sich um die Standard-Benutzernummer handelt.

Die Benutzernummern von 0 bis 99 sind für privilegierte Benutzer (Systemkennungen) reserviert. Die Benutzernummern ab 100 sind für nichtprivilegierte Benutzer vorgesehen.

Verschiedene BS2000-Benutzerkennungen mit der gleichen Benutzernummer werden auf die gleiche POSIX-Benutzerkennung abgebildet. Zwischen der BS2000-Benutzerkennung und der Benutzernummer besteht aber keine technische Abhängigkeit.

Besonders in einem UNIX-Rechnernetz ist eine eindeutige Zuordnung von BS2000-Benutzerkennung und Benutzernummer wichtig, da hier über Rechner- und Systemgrenzen hinweg konsistente Benutzeridentifikationen auf Basis der Benutzernummer vorausgesetzt werden.

## 7.4 BS2000- und POSIX-Gruppen verwalten

Da sich die Gruppenverwaltung in POSIX und im BS2000 in wesentlichen Punkten unterscheidet (siehe Abschnitt „Gruppenverwaltung“ auf Seite 40), bestehen POSIX- und BS2000-Gruppen unabhängig nebeneinander und werden getrennt verwaltet.

Der POSIX-Gruppenkatalog ist kein Bestandteil von SRPM/SECOS. Deshalb muß der Root-Berechtigte die POSIX-Gruppen separat im POSIX-Gruppenkatalog */etc/group* definieren und verwalten. Es ist seine Aufgabe, alle Veränderungen einer BS2000-Benutzerkennung (Einrichten, Gruppenwechsel, Löschen) separat im POSIX-Gruppenkatalog */etc/group* (siehe Seite 121) nachzuvollziehen.

Die Gruppennummer wird beim Anschluß des Benutzers an POSIX ohne weitere Prüfung den POSIX-Benutzerattributen entnommen. Daher liegt es im Ermessen des POSIX-Verwalters und des Root-Berechtigten, das Attribut GROUP-NUMBER und den entsprechenden POSIX-Gruppeneintrag in einer separaten Aktion aufeinander abzustimmen.

Ein BS2000-Gruppenverwalter kann die Rolle des POSIX-Verwalters für die Mitglieder seiner Gruppe übernehmen. Damit er die BS2000-Gruppenstruktur auf die POSIX-Gruppenstruktur abbilden kann, wird folgende Konvention getroffen:

„Die Gruppennummer der zur BS2000-Gruppe korrespondierenden POSIX-Gruppe ist gleich der Gruppennummer des BS2000-Gruppenverwalters.“

Ein BS2000-Gruppenverwalter hat folgende Rechte:

- Er darf seine Gruppennummer an seine BS2000-Gruppenmitglieder weitergeben. Wenn ein übergeordneter Gruppenverwalter in Vertretung des Gruppenverwalters seiner Untergruppe handelt, kann er nur dessen Gruppennummer zuweisen.
- Er kann ein BS2000-Gruppenmitglied aus der POSIX-Gruppe ausschließen, indem er ihm die Standard-Gruppennummer zuweist.

Eine darüber hinausgehende Verwaltung der POSIX-Gruppen muß zentral durch einen POSIX-Verwalter erfolgen.

### Beispiel

Die BS2000-Gruppe mit dem Gruppennamen A5 enthält folgende Benutzer:  
POSIXTST, POSIX001 und POSIX002.

Die BS2000-Gruppe mit dem Gruppennamen A7 enthält folgende Benutzer:  
MANUAL01 und MANUAL02.

Bei POSIX können dann ebenfalls Gruppen mit der Gruppennummer 5 (POSIXTST, POSIX001 und POSIX002) und der Gruppennummer 7 (MANUAL01 und MANUAL02) definiert werden.

Eine Doppelmitgliedschaft in beiden Gruppen - z.B. wenn MANUAL01 zusätzlich Mitglied der Gruppe mit der Nummer 5 werden möchte - ist aber nur möglich, wenn von der BS2000-Gruppedefinition abgewichen wird.

## POSIX-Gruppenkatalog */etc/group* der POSIX-Shell verwalten

Jeder Benutzer ist einer Benutzergruppe zugeordnet, nachdem ihm der BS2000-Systemverwalter eine numerische Gruppennummer zugewiesen hat. Dieser Gruppennummer kann der POSIX-Verwalter oder ein Root-Berechtigter im POSIX-Gruppenkatalog */etc/group* einen Gruppennamen zuordnen oder eine neue Benutzergruppe definieren. Im BS2000 gibt es kein Äquivalent zum POSIX-Gruppenkatalog */etc/group*.

Der POSIX-Gruppenkatalog */etc/group* wird bei der Erstinstallation angelegt. Er besteht aus Zeilen mit folgendem Aufbau:

<b>gruppenname: : gruppennummer : benutzerkennung[,...]</b>
---

### **gruppenname**

Name, der für diese Gruppe vergeben werden soll.

### **gruppennummer**

Numerische Gruppennummer, die im BS2000-Benutzerkatalog SYSSRPM festgelegt wurde. Dieser Gruppennummer kann mit <gruppenname> ein Gruppennname zugeordnet werden.

### **benutzerkennung**

Eine oder mehrere Benutzerkennungen, die dieser Benutzergruppe angehören sollen. Wenn mehrere Benutzerkennungen angegeben werden, müssen Sie diese durch Kommata trennen.

Eine Benutzerkennung kann mehreren Benutzergruppen zugeordnet sein.

Die Einträge müssen voneinander durch einen Doppelpunkt getrennt sein. Wenn Sie den Eintrag für das Kennwort weglassen, müssen Sie den folgenden Doppelpunkt trotzdem angeben. Die Einträge für jede Benutzergruppe müssen jeweils in einer neuen Zeile beginnen.

Der POSIX-Gruppenkatalog */etc/group* enthält nach der Erstinstallation folgende Benutzergruppen:

SYSROOT	(Gruppennummer: 0, Member : SYSROOT)
OTHER	(Gruppennummer: 1)
SYSBIN	(Gruppennummer: 2)
SYSSYS	(Gruppennummer: 3, Members: SYSROOT, SYSBIN)
MAIL	(Gruppennummer: 6, Member : SYSROOT)
TTY	(Gruppennummer: 7)
LP	(Gruppennummer: 8)
USROTHER	(Gruppennummer: 100)
DFS_STARTGID	(Gruppennummer: 2000)

## 7.5 Neue POSIX-Benutzer eintragen

Nach dem Einrichten eines neuen BS2000-Benutzers mit dem Kommando /ADD-USER (siehe Seite 127) sind die POSIX-Benutzerattribute *Benutzernummer*, *Gruppennummer*, *Login-Verzeichnis* und *Programm* mit pubsetspezifischen Standardwerten belegt (siehe Abschnitt „Standardwerte für POSIX-Benutzerattribute festlegen“ auf Seite 124). Damit der neue BS2000-Benutzer POSIX nutzen kann, muß der POSIX-Verwalter oder der BS2000-Systemverwalter die Standardwerte ändern. Das dazu erforderliche Verfahren hängt davon ab, ob auf Ihrem System das Softwareprodukt SDF-P im Einsatz ist oder nicht:

### Systeme mit SDF-P

In Systemen mit SDF-P wird die Prozedur POSADDUS angeboten, die in der Bibliothek SYSSPR.POSIX-BC.<version>.POSSTAT abgelegt ist. Diese Prozedur kann nur unter der Systemkennung TSOS ablaufen. Vor dem Start der Prozedur muß das Subsystem POSIX gestartet werden.

POSADDUS legt ein Home-Verzeichnis für den Benutzer an und trägt den Namen in das entsprechende POSIX-Benutzerattribut ein.

Die Prozedur enthält folgende Parameter:

- USERNAME (obligatorisch)
- USERNR (wahlfrei)
- GROUPNR (wahlfrei)
- DIRNAME (obligatorisch)
- PROGRAM (wahlfrei)
- ACCTRLOG (wahlfrei)

Die Parameter sind Schlüsselwortparameter. Die Parameterwerte werden als Zeichenkette übergeben.

Die Werte für DIRNAME und PROGRAM können auch in Kleinbuchstaben eingegeben werden.

Beim Parameter ACCTRLOG kann die Abrechnungsnummer für die POSIX-Nutzung über Remote Login und über NFS angegeben werden (siehe Abschnitt „Zugang von einem UNIX- oder SINIX-Rechner auf Seite 48“).

## Systeme ohne SDF-P

In Systemen ohne SDF-P müssen der POSIX-Verwalter und der Systemverwalter folgende Aufgaben durchführen:

1. POSIX-Verwalter oder Systemverwalter (im BS2000):  
Das Home-Verzeichnis, die Benutzernummer und die Gruppennummer mit dem BS2000-Kommando `/MODIFY-POSIX-USER-ATTRIBUTES` (siehe Seite 131) in den Benutzerkatalog eintragen.
2. Systemverwalter (im BS2000):  
Eventuell die Abrechnungsnummer für die POSIX-Nutzung beim Systemzugang über einen fernen Rechner mit dem BS2000-Kommando `/MODIFY-USER-ATTRIBUTES` (siehe Seite 139) ändern.
3. POSIX-Verwalter (auf der POSIX-Shell):  
Home-Verzeichnis mit dem Kommando `mkdir` einrichten; als Eigentümer wird `root` eingetragen.  
Das Kommando `mkdir` ist im Handbuch „POSIX-Kommandos“ [2] beschrieben.
4. POSIX-Verwalter (auf der POSIX-Shell):  
Neuen Benutzer als Eigentümer des Home-Verzeichnisses eintragen; dazu steht das Kommando `chown` zur Verfügung.  
Das Kommando `chown` ist im Handbuch „POSIX-Kommandos“ [2] beschrieben.
5. POSIX-Verwalter (auf POSIX-Shell):  
Dem Home-Verzeichnis mit dem Kommando `chgrp` die Gruppennummer zuweisen, die der neue Benutzer erhalten hat.  
Das Kommando `chgrp` ist im Handbuch „POSIX-Kommandos“ [2] beschrieben.



Wenn die Kennung TSOS die Root-Berechtigung besitzt, können alle Aufgaben von einer einzigen Person unter der Kennung TSOS durchgeführt werden.

## 7.6 Standardwerte für POSIX-Benutzerattribute festlegen

Die POSIX-Benutzerattribute werden beim Einrichten einer BS2000-Benutzerkennung mit den Standardwerten des angegebenen Pubsets initialisiert. Der Operandenwert \*BY-POSIX-USER-DEFAULTS bezieht sich jeweils auf die Standardwerte des angegebenen Pubsets.

Die POSIX-Benutzerattribute werden automatisch beim First-Start oder beim Versionsumstieg eingerichtet; sie werden mit fest vorgegebenen Werten initialisiert:

USER-NUMBER	=	100
GROUP-NUMBER	=	100
COMMENT	=	*NONE
DIRECTORY	=	*ROOT
PROGRAM	=	*SHELL

Allen BS2000-Benutzerkennungen außer SYSROOT wird zunächst die Standard-Benutzernummer und die Standard-Gruppennummer zugewiesen.

Die Systemkennung SYSROOT wird beim First-Start mit der fest vorgegebenen Benutzernummer 0 und der Gruppennummer 0 eingerichtet. Die Benutzernummer kann nicht geändert werden; für die Gruppennummer gibt es keine Einschränkungen.

Die Standardwerte für die Benutzerattribute können mit dem Kommando /MODIFY-POSIX-USER-DEFAULTS (siehe Seite 136) geändert werden; mit dem Kommando /SHOW-POSIX-USER-DEFAULTS (siehe Seite 153) können sie angezeigt werden.

## 7.7 Zugangsberechtigung für Benutzer eines fernen Rechners erteilen

Wenn das Softwareprodukt SECOS eingesetzt wird, kann für bereits existierende BS2000-Benutzerkennungen festgelegt werden, ob dem Benutzer eines fernen Rechners mit dem Kommando *rlogin* Zugang zum System erlaubt wird (siehe Abschnitt „Zugang zur POSIX-Shell“ auf Seite 47). Dazu steht der Operand POSIX-RLOGIN-ACCESS bei den BS2000-Kommandos /SET-LOGON-PROTECTION (siehe Seite 141) und /MODIFY-LOGON-PROTECTION (siehe Seite 129) zur Verfügung.

Mit dem Kommando /SHOW-LOGON-PROTECTION (siehe Seite 143) können die Schutzattribute angezeigt werden.

## 7.8 Abrechnungsnummer für Systemzugang über fernen Rechner eintragen

Mit den Kommandos /ADD-USER (siehe Seite 127), /MODIFY-USER-ATTRIBUTES (siehe Seite 139) und /SHOW-USER-ATTRIBUTES (siehe Seite 156) können Sie die Abrechnungsnummer beim Systemzugang über einen fernen Rechner verwalten.

## 7.9 POSIX-Benutzer löschen

Zum Löschen eines POSIX-Benutzers müssen folgende Aufgaben durchgeführt werden:

- POSIX-Verwalter:  
Individuelle Benutzernummer mit dem BS2000-Kommando /MODIFY-POSIX-USER-ATTRIBUTES (siehe Seite 131) auf den Standardwert zurücksetzen.
- Root-Berechtigter:  
Ggf. das Home-Verzeichnis mit dem POSIX-Kommando *rmdir* auf der POSIX-Shell löschen  
  
Eventuell müssen vorher die Dateien des Benutzers gelöscht oder anderen Benutzern zugewiesen werden.

Das POSIX-Kommando *rmdir* ist im Handbuch „POSIX-Kommandos“ [2] beschrieben.

## 7.10 Benutzerinformationen per Programm lesen

Der BS2000-Systemverwalter legt für jede BS2000-Benutzerkennung einen Eintrag im Benutzerkatalog an. Der Eintrag enthält u.a.:

- BS2000-Benutzerkennung, Kennwortberechtigung
- Angaben zu Systemressourcen, die der Benutzer in Anspruch nehmen kann (CPU-Zeit, Speicherplatz, ...)
- besondere Rechte des Benutzers (privilegierter Zugriff, ...)
- Daten für die Abrechnung (Accounting)

Die Daten aus dem Benutzerkatalog können mit dem Makro SRMUINF gelesen und in einen vorher festgelegten Bereich übertragen werden. Je nach Angabe werden die Daten für die Abrechnung (Accounting), die benutzerspezifischen Daten oder der gesamte Eintrag einer BS2000-Benutzerkennung aus dem Benutzerkatalog ausgegeben.

An den Operanden und Operandenwerten des Makros SRMUINF ändert sich durch POSIX nichts. Zusätzlich wird aber die POSIX-Abrechnungsnummer gekennzeichnet. Der abrechnungsspezifische Teil der Ausgabe enthält zu jeder einzelnen Abrechnungsnummer einen Indikator, der sie für die Abrechnung des Remote-Login-Systemlaufs bestimmt. Nähere Informationen zum Makro SRMUINF finden Sie im Handbuch „Makroaufrufe an den Ablaufteil“ [29].

Die Daten aus dem Benutzerkatalog können auch mit den CRTE-Makros *getlogin*, *getpwent*, *putpwent* etc. gelesen werden (siehe Handbuch „CRTE“ [7]).

## 8 BS2000-Kommandos für POSIX

In diesem Abschnitt sind alle speziellen BS2000-Kommandos für POSIX aufgeführt.

### **ADD-USER**

#### **Benutzereintrag im Benutzerkatalog erstellen**

**Anwendungsbereich:** USER-ADMINISTRATION

**Privilegierung:**           USER-ADMINISTRATION,  
                                 STD-PROCESSING

Dieses Kommando erstellt einen Eintrag im Benutzerkatalog eines Pubsets.

Die POSIX-Benutzerattribute werden implizit mit Standardwerten initialisiert. Darüber hinaus muß der BS2000-Systemverwalter für die Abrechnung eines Remote-Login-Systemlaufs eine Abrechnungsnummer bestimmen.

Folgende Benutzer dürfen dieses Kommando ausführen:

- Inhaber des Privilegs USER-ADMINISTRATION für alle BS2000-Benutzerkennungen.
- Gruppenverwalter, die mindestens das Attribut MANAGE-MEMBERS besitzen, für die BS2000-Benutzerkennungen, die ihren Gruppen zu- und untergeordnet sind.

Die folgende Syntaxdarstellung zeigt nur den POSIX-relevanten Teil des Kommandos. Das vollständige Kommando ist in den Handbüchern „SECOS“ [13] und „Kommandos“ [27] beschrieben.

**ADD-USER**

...

**,ACCOUNT-ATTRIBUTES = \*PARAMETERS(...)****\*PARAMETERS(...)****ACCOUNT = <alphanum-name 1..8>**

...

**,POSIX-RLOGIN-DEFAULT = \*NO / \*YES****ACCOUNT-ATTRIBUTES = \*PARAMETERS(...)**

Bestimmt die Abrechnungsdaten einer BS2000-Benutzerkennung.

...

**POSIX-RLOGIN-DEFAULT =**

Bestimmt, ob die angegebene Abrechnungsnummer zur Abrechnung eines Remote-Login-Systemlaufs herangezogen wird.

**POSIX-RLOGIN-DEFAULT = \*NO**

Die Abrechnungsnummer wird nicht zur Abrechnung herangezogen.

**POSIX-RLOGIN-DEFAULT = \*YES**

Die angegebene Abrechnungsnummer wird zur Abrechnung herangezogen.

*Beispiel*

Die Benutzerkennung PSXROOT wird mit der Abrechnungsnummer PSXACC eingerichtet.

```
/ADD-USER USER-ID=PSXROOT, -
/          ACCOUNT-ATTR=*PAR(ACCOUNT=PSXACC,POSIX-RLOGIN-DEFAULT=*YES)
```

## MODIFY-LOGON-PROTECTION

### Schutzattribute ändern

**Anwendungsbereich:** USER-ADMINISTRATION

**Privilegierung:** USER-ADMINISTRATION,  
STD-PROCESSING

Dieses Kommando ändert bereits bestehende Zugangsschutzattribute für BS2000-Benutzerkennungen.

Der Systemzugang über einen fernen Rechner mit dem Kommando *rlogin* wird durch die Zugangsklasse POSIX-RLOGIN-ACCESS verwaltet.

Folgende Benutzer dürfen dieses Kommando ausführen:

- Inhaber des Privilegs USER-ADMINISTRATION für alle BS2000-Benutzerkennungen.
- Gruppenverwalter, die mindestens das Attribut MANAGE-MEMBERS besitzen, für die BS2000-Benutzerkennungen, die ihren Gruppen zu- und untergeordnet sind.

Die folgende Syntaxdarstellung zeigt nur den POSIX-relevanten Teil des Kommandos. Das vollständige Kommando ist im Handbuch „SECOS“ [13] beschrieben.

#### MODIFY-LOGON-PROTECTION

...

,POSIX-RLOGIN-ACCESS = \*UNCHANGED / \*YES(...) / \*NO

\*YES(...)

    |    PASSWORD-CHECK = \*UNCHANGED / \*YES / \*NO

#### POSIX-RLOGIN-ACCESS =

Die Zugangsklassen-Attribute für den Systemzugang über einen fernen Rechner können festgelegt werden.

#### POSIX-RLOGIN-ACCESS = \*UNCHANGED

Die Zugangsklassen-Attribute für den Systemzugang über einen fernen Rechner werden nicht geändert.

**POSIX-RLOGIN-ACCESS = \*YES(...)**

Die BS2000-Benutzerkennung ist für den Systemzugang über einen fernen Rechner offen.

**PASSWORD-CHECK =**

Legt fest, ob beim Systemzugang über einen fernen Rechner das Kennwort geprüft wird.

**PASSWORD-CHECK = \*UNCHANGED**

Der bisher eingestellte Wert bleibt erhalten.

**PASSWORD-CHECK = \*YES**

Das Kennwort wird geprüft.

**PASSWORD-CHECK = \*NO**

Das Kennwort wird nicht geprüft

**POSIX-RLOGIN-ACCESS = \*NO**

Die BS2000-Benutzerkennung ist für den Systemzugang über einen fernen Rechner gesperrt.

## MODIFY-POSIX-USER-ATTRIBUTES

### POSIX-Benutzerattribute ändern

**Anwendungsbereich:** USER-ADMINISTRATION

**Privilegierung:**      POSIX-ADMINISTRATION,  
                          USER-ADMINISTRATION,  
                          STD-PROCESSING

Dieses Kommando ändert die POSIX-Benutzerattribute einer BS2000-Benutzerkennung im Benutzerkatalog des angegebenen Pubsets.

Für jede neue BS2000-Benutzerkennung werden bei ihrem Einrichten automatisch die POSIX-Benutzerattribute angelegt, die mit Standardwerten versehen sind (siehe Seite 124). Diese POSIX-Benutzerattribute können bei Bedarf geändert werden. Dazu sind folgende Benutzer berechtigt:

- Inhaber des Privilegs POSIX-ADMINISTRATION oder USER-ADMINISTRATION für alle BS2000-Benutzerkennungen auf allen Pubsets.
- Gruppenverwalter für die ihnen unterstellten Gruppen- und Untergruppenmitglieder auf dem von ihnen verwalteten Pubset. Für einen Gruppenverwalter gelten aber folgende Einschränkungen:
  - Seine Autorisierung ADM-AUTHORITY bestimmt die POSIX-Benutzerattribute, zu deren Verwaltung er berechtigt ist.
  - Der Wertebereich der POSIX-Benutzerattribute ist für ihn eingeschränkt.

Näheres dazu steht beim entsprechenden Operanden in der Operandenbeschreibung.

#### MODIFY-POSIX-USER-ATTRIBUTES

**USER-IDENTIFICATION** = <name 1..8>

**,PUBSET** = \*HOME / <catid 1..4>

**,USER-NUMBER** = \*UNCHANGED / \*BY-POSIX-USER-DEFAULTS / \*HOME / <integer 0..60002>

**,GROUP-NUMBER** = \*UNCHANGED / \*BY-POSIX-USER-DEFAULTS / \*GROUP-ADMINISTRATOR /  
<integer 0..60002>

**,COMMENT** = \*UNCHANGED / \*BY-POSIX-USER-DEFAULTS / \*NONE / <c-string 1..255 with-low>

**,DIRECTORY** = \*UNCHANGED / \*BY-POSIX-USER-DEFAULTS / \*ROOT /  
<posix-pathname 1..1023 without-wild>

**,PROGRAM** = \*UNCHANGED / \*BY-POSIX-USER-DEFAULTS / \*SHELL /  
<posix-pathname 1..1023 without-wild>

**USER-IDENTIFICATION = <name 1..8>**

BS2000-Benutzerkennung, deren POSIX-Benutzerattribute geändert werden sollen.

**PUBSET =**

Pubset, in dessen Benutzerkatalog die POSIX-Benutzerattribute geändert werden sollen.

**PUBSET = \*HOME**

Die Änderung erfolgt auf dem Home-Pubset.

**PUBSET = <catid 1..4>**

Die Änderung erfolgt auf dem Pubset mit der angegebenen Katalogkennung.

**USER-NUMBER =**

Die Benutzernummer, die beim Einrichten einer BS2000-Benutzerkennung automatisch vergeben wird, kann geändert werden.

Das Attribut USER-NUMBER ist sicherheitsrelevant, da die Benutzernummer die Privilegierung ausdrückt und den Eigentümer einer Datei bestimmt.

Der Gruppenverwalter kann die Benutzernummer nur ändern, wenn er mindestens das Gruppenverwalterrecht MANAGE-MEMBERS besitzt. Für ihn ist aber der Wertebereich eingeschränkt:

- Er kann nicht die Benutzernummer 0 vergeben, d.h. die Root-Berechtigung.
- Er kann nur die Standard-Benutzernummer ändern.
- Er kann nur Benutzernummern vergeben, die größer als die Standard-Benutzernummer sind.
- Er kann Benutzernummern nicht mehrfach vergeben.
- Er kann auf einem Daten-Pubset nur die Benutzernummer der gleichnamigen BS2000-Benutzerkennung auf dem Home-Pubset zuweisen.

**USER-NUMBER = \*UNCHANGED**

Die Benutzernummer wird nicht geändert.

**USER-NUMBER = \*BY-POSIX-USER-DEFAULTS**

Die Benutzernummer erhält den entsprechenden Standardwert, der im Benutzerkatalog des angegebenen Pubsets eingetragen ist.

**USER-NUMBER = \*HOME**

Die Benutzernummer der gleichnamigen BS2000-Benutzerkennung auf dem Home-Pubset wird übernommen.

Dieser Wert ist nur von Bedeutung, wenn die Benutzernummer auf einem Daten-Pubset geändert wird. Auf dem Home-Pubset ist diese Angabe redundant.

**USER-NUMBER = <integer 0..60002>**

Die Benutzernummer erhält den angegebenen Wert.

**GROUP-NUMBER =**

Die Gruppennummer, die beim Einrichten einer BS2000-Benutzerkennung automatisch vergeben wird, kann geändert werden.

Das Attribut GROUP-NUMBER ist sicherheitsrelevant, da POSIX beim Login nicht die Zulässigkeit der Kombination BS2000-Benutzerkennung und -Gruppe gegen den POSIX-Gruppenkatalog prüft.

Der Gruppenverwalter kann die Gruppennummer nur ändern, wenn er das Gruppenverwalterrecht MANAGE-MEMBERS besitzt. Für ihn ist aber der Wertebereich eingeschränkt:

- Er kann nur die Gruppennummer vergeben, die der Gruppenverwalter der BS2000-Benutzergruppe besitzt, deren Mitglied die BS2000-Benutzerkennung ist, oder die Standard-Gruppennummer.
- Er kann für seine eigene BS2000-Benutzerkennung keine andere Gruppennummer vergeben.

**GROUP-NUMBER = \*UNCHANGED**

Die Gruppennummer wird nicht geändert.

**GROUP-NUMBER = \*BY-POSIX-USER-DEFAULTS**

Die Gruppennummer erhält den entsprechenden Standardwert, der im Benutzerkatalog des angegebenen Pubsets eingetragen ist.

**GROUP-NUMBER = \*GROUP-ADMINISTRATOR**

Es wird die Gruppennummer vergeben, die der Gruppenverwalter der BS2000-Benutzergruppe besitzt, deren Mitglied die BS2000-Benutzerkennung ist.

**GROUP-NUMBER = <integer 0..60002>**

Die Gruppennummer erhält den angegebenen Wert.

**COMMENT =**

Der Kommentar kann geändert werden. Nach eigenem Ermessen können nähere Angaben zum Eigentümer der BS2000-Benutzerkennung gemacht werden.

**COMMENT = \*UNCHANGED**

Der Kommentar wird nicht geändert.

**COMMENT = \*BY-POSIX-USER-DEFAULTS**

Der Wert des entsprechenden POSIX-Standardattributs wird übernommen, das im Benutzerkatalog des angegebenen Pubsets eingetragen ist.

**COMMENT = \*NONE**

Es wird kein Kommentar eingetragen.

**COMMENT = <c-string 1..255 with-low>**

Der angegebene Kommentar wird eingetragen.

**DIRECTORY =**

Der absolute Pfadname zum Login-Verzeichnis des Benutzers kann geändert werden.

Dieses Attribut ist nicht sicherheitsrelevant, da es nur den Inhalt der Shell-Variablen *HOME* und den Anfangswert des Arbeitsverzeichnisses bestimmt. Die Schutzattribute von Dateien und Dateiverzeichnissen können damit nicht umgangen werden.

**DIRECTORY = UNCHANGED**

Der absolute Pfadname wird nicht geändert.

**DIRECTORY = \*BY-POSIX-USER-DEFAULTS**

Der Wert des entsprechenden POSIX-Standardattributs wird übernommen, das im Benutzerkatalog des angegebenen Pubsets eingetragen ist.

**DIRECTORY = \*ROOT**

Das Root-Verzeichnis "/" wird zugeordnet.

**DIRECTORY = <posix-pathname 1..1023 without-wild>**

Der angegebene Pfadname wird übernommen.

**PROGRAM =**

Das Programm kann geändert werden, das nach dem Kommando *rlogin* bzw. nach dem Aufruf des Kommandos START-POSIX-SHELL gestartet wird.

Dieses Attribut ist nicht sicherheitsrelevant, da nur solche Programme gestartet werden können, die der Benutzer ausführen darf.

**PROGRAM = UNCHANGED**

Das Programm wird nicht geändert.

**PROGRAM = \*BY-POSIX-USER-DEFAULTS**

Der Wert des entsprechenden POSIX-Standardattributs wird übernommen, das im Benutzerkatalog des angegebenen Pubsets eingetragen ist.

**PROGRAM = \*SHELL**

Die POSIX-Shell wird gestartet.

**PROGRAM = <posix-pathname 1..1023 without-wild>**

Das angegebene Programm wird gestartet.

**Kommando-Returncodes**

<b>(SC2)</b>	<b>SC1</b>	<b>Maincode</b>	<b>Bedeutung</b>
2	0	CMD0001	Kommando fehlerfrei ausgeführt
	0	SRM6001	Kommando mit Warnung ausgeführt
	32	SRM6020	Kommando wegen eines Systemfehlers abgewiesen
	130	SRM6030	Kommando wegen Ressourcenmangel abgewiesen
	64	SRM6040	Kommando mit Fehlermeldung abgewiesen

*Beispiele*

Der Benutzerkennung POSIXTST soll die Benutzernummer 107 und die Gruppennummer 66 zugeordnet werden. Das Login-Verzeichnis (Home-Verzeichnis) soll */home/posixtst* heißen. Nach dem POSIX-Login soll die Bourne-Shell aufgerufen werden. Der Kommentar soll „posix user“ lauten.

```
/MODIFY-POSIX-USER-ATTRIBUTES USER-ID=POSIXTST, -
/                               USER-NUMBER=107, -
/                               GROUP-NUMBER=66, -
/                               DIRECTORY=/home/posixtst, -
/                               COMMENT='posix user'
```

Die Benutzerkennung PSXROOT soll die Root-Berechtigung erhalten. Als Home-Verzeichnis soll */home/psxroot* eingetragen werden.

```
/MODIFY-POSIX-USER-ATTRIBUTES USER-ID=PSXROOT, -
/                               USER-NUMBER=0, -
/                               GROUP-NUMBER=0, -
/                               DIRECTORY=/home/psxroot
```

## MODIFY-POSIX-USER-DEFAULTS

### Standardwerte für POSIX-Benutzerattribute ändern

**Anwendungsbereich:** USER-ADMINISTRATION

**Privilegierung:**      POSIX-ADMINISTRATION,  
                          USER-ADMINISTRATION,  
                          STD-PROCESSING

Dieses Kommando ändert die POSIX-Standardattribute im Benutzerkatalog des angegebenen Pubsets. Folgende Benutzer dürfen es ausführen:

- Inhaber des Privilegs POSIX-ADMINISTRATION oder USER-ADMINISTRATION für alle Pubsets.
- Gruppenverwalter der Gruppe \*UNIVERSAL auf dem von ihnen verwalteten Pubset.

Die POSIX-Standardattribute werden beim Anlegen eines neuen Benutzers (mit ADD-USER) verwendet.

#### MODIFY-POSIX-USER-DEFAULTS

```
PUBSET = *HOME / <catid 1..4>
,USER-NUMBER = *UNCHANGED / <integer 0..60002>
,GROUP-NUMBER = *UNCHANGED / <integer 0..60002>
,COMMENT = *UNCHANGED / *NONE / <c-string 1..255 with-low>
,DIRECTORY = *UNCHANGED / *ROOT / <posix-pathname 1..1023 without-wild>
,PROGRAM = *UNCHANGED / *SHELL / <posix-pathname 1..1023 without-wild>
```

#### PUBSET =

Pubset, in dessen Benutzerkatalog die POSIX-Standardattribute geändert werden sollen.

#### PUBSET = \*HOME

Die POSIX-Standardattribute werden im Benutzerkatalog des Home-Pubsets geändert.

#### PUBSET = <catid 1..4>

Die POSIX-Standardattribute werden im Benutzerkatalog des angegebenen Pubsets geändert.

**USER-NUMBER =**

Die Benutzernummer kann geändert werden.

**USER-NUMBER = \*UNCHANGED**

Die Benutzernummer wird nicht geändert.

**USER-NUMBER = <integer 0..60002>**

Die Benutzernummer erhält den angegebenen Wert.

**GROUP-NUMBER =**

Die Gruppennummer kann geändert werden.

**GROUP-NUMBER = \*UNCHANGED**

Die Gruppennummer wird nicht geändert.

**GROUP-NUMBER = <integer 0..60002>**

Die Gruppennummer erhält den angegebenen Wert.

**COMMENT =**

Der Kommentar kann geändert werden.

**COMMENT = \*UNCHANGED**

Der Kommentar wird nicht geändert.

**COMMENT = \*NONE**

Es wird kein Kommentar eingetragen.

**COMMENT = <c-string 1..255 with-low>**

Der angegebene Kommentar wird eingetragen.

**DIRECTORY =**

Der absolute Pfadname zum Login-Verzeichnis des Benutzers kann geändert werden.

**DIRECTORY = \*UNCHANGED**

Der absolute Pfadname wird nicht geändert.

**DIRECTORY = \*ROOT**

Es wird ins Root-Verzeichnis gewechselt.

**DIRECTORY = <posix-pathname 1..1023 without-wild>**

Es wird zum angegebenen Pfadnamen gewechselt.

**PROGRAM =**

Das Programm, das nach der Anmeldung des Benutzers gestartet wird, kann geändert werden.

**PROGRAM = UNCHANGED**

Das Programm wird nicht geändert.

**PROGRAM = \*SHELL**

Die standardmäßige POSIX-Shell wird gestartet.

**PROGRAM = <posix-pathname 1..1023 without-wild>**

Das angegebene Programm wird gestartet.

**Kommando-Returncodes**

<b>(SC2)</b>	<b>SC1</b>	<b>Maincode</b>	<b>Bedeutung</b>
2	0	CMD0001	Kommando fehlerfrei ausgeführt
	0	SRM6001	Kommando mit Warnung ausgeführt
	32	SRM6020	Kommando wegen eines Systemfehlers abgewiesen
	64	SRM6040	Kommando mit Fehlermeldung abgewiesen
	130	SRM6030	Kommando wegen Ressourcenmangel abgewiesen

## MODIFY-USER-ATTRIBUTES

### Katalogeintrag eines Benutzers ändern

**Anwendungsbereich:** ACCOUNTING,  
USER-ADMINISTRATION

**Privilegierung:** USER-ADMINISTRATION,  
STD-PROCESSING

Dieses Kommando ändert die Attribute einer BS2000-Benutzerkennung im Benutzerkatalog. Folgende Benutzer dürfen es ausführen:

- Inhaber des Privilegs USER-ADMINISTRATION für alle BS2000-Benutzerkennungen.
- Gruppenverwalter für die BS2000-Benutzerkennungen, die ihren Gruppen zu- und untergeordnet sind.

Die folgende Syntaxdarstellung zeigt nur den POSIX-relevanten Teil des Kommandos. Das vollständige Kommando ist in den Handbüchern „SECOS“ [13] und „Kommandos“ [27] beschrieben.

#### MODIFY-USER-ATTRIBUTES

...

,ACCOUNT-ATTRIBUTES = \*UNCHANGED / \*ADD(...) / \*MODIFY(...) / \*REMOVE(...)

\*ADD(...)

    ACCOUNT = <alphanum-name 1..8>

...

,POSIX-RLOGIN-DEFAULT = \*NO / \*YES

\*MODIFY(...)

    ACCOUNT = <alphanum-name 1..8>

...

,POSIX-RLOGIN-DEFAULT = \*UNCHANGED / \*NO / \*YES

\*REMOVE(...)

**ACCOUNT-ATTRIBUTES = ... / \*ADD(...) / \*MODIFY(...) / ...**

Legt die Abrechnungsdaten einer BS2000-Benutzerkennung fest.

**ACCOUNT-ATTRIBUTES = \*ADD(...)**

Eine neue Abrechnungsnummer und spezifische Attribute werden für die BS2000-Benutzerkennung eingetragen.

**ACCOUNT = <alphanum-name 1..8>**

Abrechnungsnummer der BS2000-Benutzerkennung, die in den Benutzerkatalog aufgenommen wird und auf die sich die folgenden Angaben beziehen.

...

**POSIX-RLOGIN-DEFAULT =**

Legt fest, ob die angegebene Abrechnungsnummer zur Abrechnung des Remote-Login-Systemlaufs herangezogen wird.

**POSIX-RLOGIN-DEFAULT = \*NO**

Die angegebene Abrechnungsnummer wird nicht zur Abrechnung herangezogen.

**POSIX-RLOGIN-DEFAULT = \*YES**

Die angegebene Abrechnungsnummer wird zur Abrechnung herangezogen.

**ACCOUNT-ATTRIBUTES = \*MODIFY(...)**

Die Attribute einer eingetragenen Abrechnungsnummer der BS2000-Benutzerkennung werden geändert.

**ACCOUNT = <alphanum-name 1..8>**

Abrechnungsnummer der BS2000-Benutzerkennung, für die die nachfolgenden Werte im Benutzerkatalog geändert werden.

...

**POSIX-RLOGIN-DEFAULT =**

Legt fest, ob die zu ändernde Abrechnungsnummer zur Abrechnung des Remote-Login-Systemlaufs herangezogen wird.

**POSIX-RLOGIN-DEFAULT = \*UNCHANGED**

Der bisher eingestellte Wert bleibt erhalten.

**POSIX-RLOGIN-DEFAULT = \*NO**

Die Abrechnungsnummer wird nicht zur Abrechnung herangezogen.

**POSIX-RLOGIN-DEFAULT = \*YES**

Die Abrechnungsnummer wird zur Abrechnung herangezogen.



Innerhalb einer BS2000-Benutzerkennung ist die Abrechnungsnummer für Remote Login eindeutig. Die Benutzerverwaltung führt automatisch einen Abgleich mit den vorhandenen Abrechnungsnummern durch.

Die Abrechnungsnummer für Remote Login kann auch für die Abrechnung eines BS2000-Systemlaufs angegeben werden. Dadurch kann eine BS2000-Benutzerkennung mit einer einzigen Abrechnungsnummer auskommen.

Wenn keine Abrechnungsnummer für Remote Login angegeben ist, wird der Systemzugang über Remote Login abgewiesen, außer wenn es sich um die Benutzerkennung TSOS handelt.

## SET-LOGON-PROTECTION

### Schutzattribute vereinbaren

**Anwendungsbereich:** USER-ADMINISTRATION

**Privilegierung:** USER-ADMINISTRATION,  
STD-PROCESSING

Dieses Kommando vereinbart Zugangsschutzattribute für existierende BS2000-Benutzerkennungen.

Der Systemzugang für den Benutzer eines fernen Rechners mit dem Kommando *rlogin* wird durch die Zugangsklasse POSIX-RLOGIN-ACCESS verwaltet.

Folgende Benutzer dürfen dieses Kommando ausführen:

- Inhaber des Privilegs USER-ADMINISTRATION für alle BS2000-Benutzerkennungen.
- Gruppenverwalter, die mindestens das Attribut MANAGE-MEMBERS besitzen, für die BS2000-Benutzerkennungen, die ihren Gruppen zu- und untergeordnet sind.

Die folgende Syntaxdarstellung zeigt nur den POSIX-relevanten Teil des Kommandos. Das vollständige Kommando finden Sie im Handbuch „SECOS“ [13].

#### SET-LOGON-PROTECTION

...

,POSIX-RLOGIN-ACCESS = \*YES (...) / \*NO

\*YES(...)

    |    PASSWORD-CHECK = \*YES / \*NO

#### POSIX-RLOGIN-ACCESS =

Die Zugangsklassen-Attribute für den Systemzugang über einen fernen Rechner können festgelegt werden.

**POSIX-RLOGIN-ACCESS = \*YES(...)**

Die BS2000-Benutzerkennung ist für den Systemzugang über einen fernen Rechner offen.

**PASSWORD-CHECK =**

Legt fest, ob beim Systemzugang über einen fernen Rechner das Kennwort geprüft wird.

**PASSWORD-CHECK = \*YES**

Das Kennwort wird geprüft.

**PASSWORD-CHECK = \*NO**

Das Kennwort wird nicht geprüft.

**POSIX-RLOGIN-ACCESS = \*NO**

Die BS2000-Benutzerkennung ist für den Systemzugang über einen fernen Rechner gesperrt.

*Beispiel*

Die Benutzerkennung PSXROOT wird für den Systemzugang über einen fernen Rechner zugelassen:

```
/SET-LOGON-PROTECTION USER-ID=PSXROOT,POSIX-RLOGIN-ACCESS=*YES
```

## SHOW-LOGON-PROTECTION

### Schutzattribute anzeigen

**Anwendungsbereich:** USER-ADMINISTRATION

**Privilegierung:** USER-ADMINISTRATION,  
STD-PROCESSING

Dieses Kommando zeigt die Zugangsschutzattribute von BS2000-Benutzerkennungen an. Die Ausgabe enthält u.a. die Attribute der Zugangsklasse POSIX-RLOGIN-ACCESS.

Folgende Benutzer dürfen dieses Kommando ausführen:

- Inhaber des Privilegs USER-ADMINISTRATION für alle BS2000-Benutzerkennungen.
- Gruppenverwalter, die mindestens das Attribut MANAGE-MEMBERS besitzen, für die BS2000-Benutzerkennungen, die ihren Gruppen zu- und untergeordnet sind.
- jeder Benutzer für seine eigene BS2000-Benutzerkennung.

An der Benutzeroberfläche des Kommandos SHOW-LOGON-PROTECTION ändert sich durch POSIX nichts. Das vollständige Kommando finden Sie im Handbuch „SECOS“ [13].

#### S-Variablen

Die Attribute der Zugangsklasse POSIX-RLOGIN-ACCESS werden in folgenden S-Variablen abgelegt:

Ausgabe-Information	Name der S-Variablen	T	Inhalt	Bedingung
Zugangskontrolle beim POSIX-Zugang über rlogin aktiv?	var(*LIST).POSIX-RLOG.ACCESS	S	*NO *YES	
Kennwortprüfung beim POSIX-Zugang über rlogin aktiv?	var(*LIST).POSIX-RLOG.PASS-CHECK	S	*NO *YES	

Nähere Informationen zu S-Variablen finden Sie im Handbuch „Kommandos Band 4“ [27].

## SHOW-POSIX-STATUS

### POSIX-Status anzeigen

**Anwendungsbereich:** SYSTEM-MANAGEMENT

**Privilegierung:** SUBSYSTEM-MANAGEMENT

Dieses Kommando gibt den Zustand von POSIX aus. Zur erfolgreichen Kommandoausführung muß das kostenpflichtige Subsystem SDF-P geladen sein.

SHOW-POSIX-STATUS

Einer der folgenden Zustände wird ausgegeben:

POSIX-STATUS	Bedeutung
*AVAILABLE	POSIX ist für Anwendungen freigegeben.
*IN-CREATE	Das Subsystem POSIX wird gestartet.
*IN-DELETE	Das Subsystem POSIX wird beendet.
*NOT-ACCESSIBLE	Das Subsystem POSIX ist gestartet, aber nicht für Anwendungen freigegeben.
*NOT-AVAILABLE	Das Subsystem POSIX ist nicht geladen.
*UNKNOWN	Der Zustand konnte nicht ermittelt werden.

#### Kommando-Returncodes

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Kommando fehlerfrei ausgeführt

## SHOW-POSIX-USER-ATTRIBUTES

### POSIX-Benutzerattribute anzeigen

**Anwendungsbereich:** USER-ADMINISTRATION

**Privilegierung:** POSIX-ADMINISTRATION,  
USER-ADMINISTRATION,  
STD-PROCESSING

Dieses Kommando zeigt die POSIX-Benutzerattribute einer BS2000-Benutzererkennung an, die im Benutzerkatalog des angegebenen Pubsets eingetragen sind. Folgende Benutzer dürfen es ausführen:

- Inhaber des Privilegs POSIX-ADMINISTRATION oder USER-ADMINISTRATION für alle BS2000-Benutzererkennungen auf allen Pubsets.
- Gruppenverwalter für die Gruppen- und Untergruppenmitglieder, die ihnen unterstellt sind, auf dem von ihnen verwalteten Pubset.
- jeder Benutzer für seine eigene BS2000-Benutzererkennung.

#### SHOW-POSIX-USER-ATTRIBUTES

**USER-IDENTIFICATION** = **\*OWN** / **\*ALL** / list-poss(20): <name 1..8>

,**PUBSET** = **\*HOME** / **\*ALL** / list-poss(20): <catid 1..4>

,**SELECT** = **\*ALL** / **\*BY-ATTRIBUTES**(...)

**\*BY-ATTRIBUTES**(...)

**USER-NUMBER** = **\*ANY** / **\*BY-POSIX-USER-DEFAULTS** / **\*OWN** / <integer 0..60002>

    ,b**GROUP-NUMBER** = **\*ANY** / **\*BY-POSIX-USER-DEFAULTS** / **\*OWN** / <integer 0..60002>

    ,b**COMMENT** = **\*ANY** / **\*BY-POSIX-USER-DEFAULTS** / **\*NONE** / <c-string 1..255 with-low>

    ,b**DIRECTORY** = **\*ANY** / **\*BY-POSIX-USER-DEFAULTS** / **\*ROOT** /  
        <posix-pathname 1..1023 without-wild>

    ,b**PROGRAM** = **\*ANY** / **\*BY-POSIX-USER-DEFAULTS** / **\*SHELL** /  
        <posix-pathname 1..1023 without-wild>

,**INFORMATION** = **\*ALL** / **\*USER-LIST**

,**OUTPUT** = list-poss(2): **\*SYSOUT** / **\*SYSLST**(...)

**\*SYSLST**(...)

    ,b**SYSLST-NUMBER** = **\*STD** / <integer 1..99>

    ,b**INES-PER-PAGE** = **64** / <integer 20..255>

**USER-IDENTIFICATION =**

BS2000-Benutzerkennungen, deren POSIX-Benutzerattribute angezeigt werden sollen.

**USER-IDENTIFICATION = OWN**

Die POSIX-Benutzerattribute der eigenen BS2000-Benutzerkennung werden angezeigt, die im Benutzerkatalog des angegebenen Pubsets eingetragen sind.

**USER-IDENTIFICATION = ALL**

Die POSIX-Benutzerattribute aller BS2000-Benutzerkennungen, zu deren Kenntnisnahme der Aufrufer berechtigt ist, werden angezeigt.

**USER-IDENTIFICATION = list-poss(20): <name 1..8>**

Die POSIX-Benutzerattribute der angegebenen Kennung werden angezeigt.

**PUBSET =**

Bestimmt das Pubset, aus dessen Benutzerkatalog die POSIX-Benutzerattribute angezeigt werden sollen.

**PUBSET = HOME**

Die POSIX-Benutzerattribute des Home-Pubsets werden angezeigt.

**PUBSET = ALL**

Die POSIX-Benutzerattribute aller Pubsets, die zum Zeitpunkt der Kommandoeingabe verfügbar sind, werden angezeigt.

**PUBSET = list-poss(20): <catid 1..4>**

Die POSIX-Benutzerattribute des angegebenen Pubsets werden angezeigt.

**SELECT =**

Die BS2000-Benutzerkennungen werden anhand ihrer POSIX-Benutzerattribute ausgewählt.

**SELECT = ALL**

Die BS2000-Benutzerkennungen werden unabhängig von ihren POSIX-Benutzerattributen ausgewählt.

**SELECT = \*BY-ATTRIBUTES(...)**

Die BS2000-Benutzerkennungen werden abhängig von ihren POSIX-Benutzerattributen ausgewählt. Wenn mehr als ein POSIX-Benutzerattribut angegeben ist, erfolgt die Auswahl durch „und“-Verknüpfung.

**USER-NUMBER =**

Die BS2000-Benutzerkennungen werden anhand ihrer Benutzernummer ausgewählt.

**USER-NUMBER = \*ANY**

Die BS2000-Benutzerkennungen werden unabhängig von ihrer Benutzernummer ausgewählt.

**USER-NUMBER = \*BY-POSIX-USER-DEFAULTS**

Nur die BS2000-Benutzerkennungen werden ausgewählt, für die als Benutzernummer der entsprechende Standardwert im Benutzerkatalog des angegebenen Pubsets eingetragen ist.

**USER-NUMBER = \*OWN**

Nur die BS2000-Benutzerkennungen werden ausgewählt, die dieselbe Benutzernummer wie der Aufrufer im Benutzerkatalog des angegebenen Pubsets eingetragen haben.

**USER-NUMBER = <integer 0..60002>**

Nur die BS2000-Benutzerkennungen werden ausgewählt, die die angegebene Benutzernummer im Benutzerkatalog des angegebenen Pubsets eingetragen haben.

**GROUP-NUMBER =**

Die BS2000-Benutzerkennungen werden anhand ihrer Gruppennummer ausgewählt.

**GROUP-NUMBER = \*ANY**

Die BS2000-Benutzerkennungen werden unabhängig von ihrer Gruppennummer ausgewählt.

**GROUP-NUMBER = \*BY-POSIX-USER-DEFAULTS**

Nur die BS2000-Benutzerkennungen werden ausgewählt, für die als Gruppennummer der entsprechende Standardwert im Benutzerkatalog des angegebenen Pubsets eingetragen ist.

**GROUP-NUMBER = \*OWN**

Nur die BS2000-Benutzerkennungen werden ausgewählt, die dieselbe Gruppennummer wie der Aufrufer im Benutzerkatalog des angegebenen Pubsets eingetragen haben.

**GROUP-NUMBER = <integer 0..60002>**

Nur die BS2000-Benutzerkennungen werden ausgewählt, die die angegebene Gruppennummer im Benutzerkatalog des angegebenen Pubsets eingetragen haben.

**COMMENT =**

Die BS2000-Benutzerkennungen werden anhand ihres Kommentars ausgewählt.

**COMMENT = \*ANY**

Die BS2000-Benutzerkennungen werden unabhängig von ihrem Kommentar ausgewählt.

**COMMENT = \*BY-POSIX-USER-DEFAULTS**

Nur die BS2000-Benutzerkennungen werden ausgewählt, für die als Kommentar der entsprechende Standardwert im Benutzerkatalog des angegebenen Pubsets eingetragen ist.

**COMMENT = \*NONE**

Nur die BS2000-Benutzerkennungen mit einem leeren Kommentar werden ausgewählt.

**COMMENT = <c-string 1..255 with-low>**

Nur die BS2000-Benutzerkennungen mit dem angegebenen Kommentar werden ausgewählt.

**DIRECTORY =**

Die BS2000-Benutzerkennungen werden anhand ihres Login-Verzeichnisses ausgewählt.

**DIRECTORY = \*ANY**

Die BS2000-Benutzerkennungen werden unabhängig von ihrem Login-Verzeichnis ausgewählt.

**DIRECTORY= \*BY-POSIX-USER-DEFAULTS**

Nur die BS2000-Benutzerkennungen werden ausgewählt, für die als Login-Verzeichnis der entsprechende Standardwert im Benutzerkatalog des angegebenen Pubsets eingetragen ist.

**DIRECTORY = \*ROOT**

Nur die BS2000-Benutzerkennungen werden ausgewählt, die als Login-Verzeichnis das Root-Verzeichnis eingetragen haben.

**DIRECTORY = <posix-pathname 1..1023 without-wild>**

Nur die BS2000-Benutzerkennungen mit dem angegebenen Login-Verzeichnis werden ausgewählt.

**PROGRAM =**

Die BS2000-Benutzerkennungen werden anhand ihres Programmnamens ausgewählt.

**PROGRAM = \*ANY**

Die BS2000-Benutzerkennungen werden unabhängig vom Programmnamen ausgewählt.

**PROGRAM = \*BY-POSIX-USER-DEFAULTS**

Nur die BS2000-Benutzerkennungen werden ausgewählt, für die als Programmname der entsprechende Standardwert im Benutzerkatalog des angegebenen Pubsets eingetragen ist.

**PROGRAM = \*SHELL**

Nur die BS2000-Benutzerkennungen werden ausgewählt, die als Programmnamen \*SHELL eingetragen haben.

**PROGRAM = <posix-pathname 1..1023 without-wild>**

Nur die BS2000-Benutzerkennungen mit dem angegebenen Programmnamen werden ausgewählt.

**INFORMATION =**

Umfang der Informationsausgabe.

**INFORMATION = \*ALL**

Alle POSIX-Benutzerattribute einer BS2000-Benutzerkennung werden angezeigt (siehe Beispiel 1 auf Seite 150).

**INFORMATION = \*USER-LIST**

Eine Liste der BS2000-Benutzerkennungen ohne POSIX-Benutzerattribute wird angezeigt (siehe Beispiel 2 auf Seite 151).

**OUTPUT =**

Systemdatei für die Ausgabe der Information.

**OUTPUT = \*SYSOUT**

Die Information wird in die Systemdatei SYSOUT ausgegeben.

**OUTPUT = \*SYSLST(...)**

Die Information wird in die Systemdatei SYSLST ausgegeben.

**SYSLST-NUMBER =**

Bestimmt die SYSLST-Nummer.

**SYSLST-NUMBER = \*STD**

Bestimmt die Standard-SYSLST-Ausgabe.

**SYSLST-NUMBER = <integer 1..99>**

Bestimmt die angegebene SYSLST-Nummer.

**LINES-PER-PAGE =**

Gibt die Zeilenzahl pro Seite an.

**LINES-PER-PAGE = 64**

Standardmäßig werden 64 Zeilen pro Seite gedruckt.

**LINES-PER-PAGE = <integer 20..255>**

Die angegebene Zeilenzahl wird pro Seite gedruckt.



Ein Benutzer ohne Verwaltungstätigkeit erhält bis auf zwei Ausnahmen nur Informationen über seine eigene BS2000-Benutzerkennung:

INFORMATION=\*USER-LIST,SELECT=\*BY-ATTRIBUTES(USER-NUMBER=\*OWN)

Bei dieser Angabe erfährt der Benutzer auch die Identität der Benutzer, die dieselbe Benutzernummer wie er haben, wenn diese Benutzernummer ungleich der Standard-Benutzernummer ist.

INFORMATION=\*USER-LIST,SELECT=\*BY-ATTRIBUTES(GROUP-NUMBER=\*OWN)

Bei dieser Angabe erfährt der Benutzer auch die Identität der Mitglieder seiner POSIX-Gruppe, wenn diese POSIX-Gruppe ungleich der Standardgruppe ist.

Bei INFORMATION=\*ALL werden die Benutzernummer und die Gruppennummer gekennzeichnet, wenn der entsprechende Standardwert, der im Benutzerkatalog des angegebenen Pubsets eingetragen ist, zugewiesen ist (SHOW-Ausgabe mit „(DEFAULT)“ bzw. S-Variablen mit dem Suffix „-DEF“).

### Kommando-Returncodes

<b>(SC2)</b>	<b>SC1</b>	<b>Maincode</b>	<b>Bedeutung</b>
2	0	CMD0001	Kommando fehlerfrei ausgeführt
	0	SRM6001	Kommando mit Warnung ausgeführt
	32	CMD2009	Fehler beim Erzeugen der Ausgabe-Variablen
	32	SRM6020	Kommando wegen eines Systemfehlers abgewiesen
	64	OPS0002	K2-Unterbrechung bei Ausgabe in S-Variable
	64	SRM6040	Kommando mit Fehlermeldung abgewiesen
	130	OPS0001	Ressourcenmangel bei Ausgabe in S-Variable
	130	SRM6030	Kommando wegen Ressourcenmangel abgewiesen

### Beispiel 1

```
/SHOW-POSIX-USER-ATTRIBUTES USER-IDENTIFICATION=EXAMPLE1,PUBSET=A
```

### Ausgabe:

```
/SHOW-POSIX-USER-ATTRIBUTES USER-IDENTIFICATION=EXAMPLES,PUBSET=A
POSIX-USER-ATTRIBUTES --- PUBSET A 1998-02-17 17:19:48
```

```
-----  

USER-ID           EXAMPLES          PUBSET   A  

USER-NUMBER       632                

GROUP-NUMBER      632                

COMMENT          S. Nobody, Mch-P, Tel.: 12345  

DIRECTORY        /home/examples  

PROGRAM          *SHELL  

-----
```

```
POSIX-USER-ATTRIBUTES          END OF DISPLAY
```

*Beispiel 2*

```
/SHOW-POSIX-USER-ATTRIBUTES USER-ID=*ALL,PUBSET=A, -
/                               INFORMATION=*USER-LIST
```

*Ausgabe:*

```
/SHOW-POSIX-USER-ATTRIBUTES USER-ID=*ALL,PUBSET=A,INFORMATION=*USER-LIST
POSIX-USER-LIST      --- PUBSET A          1998-02-17 17:23:12
-----
EXAMPLES   SERVICE    SYSAUDIT   SYSDUMP   SYSGEN    SYSHSMS   SYSNAC   SYSPRIV
SYSROOT   SYSSNAP   SYSSPOOL   SYSUSER   TSOS
-----
POSIX-USER-LIST          END OF DISPLAY
```

**S-Variablen**

Mit dem Operanden INFORMATION des Kommandos wird festgelegt, welche S-Variablen mit Werten versorgt werden. Folgende Angaben sind für INFORMATION möglich:

Schreibweise im Kommando	gekürzte Schreibweise in Tabelle
INFORMATION = *ALL	INF=ALL
INFORMATION = *USER-LIST	INF=U-LIST

Bitte beachten Sie, daß S-Variablen nur erzeugt werden, wenn die entsprechenden Bedingungen (siehe Spalte „Bedingung“) gültig sind.

Ausgabe-Information	Name der S-Variablen	T	Inhalt	Bedingung
Kommentar, anhand dessen die BS2000-Benutzerkennung ausgewählt wird	var(*LIST).COMMENT	S	*NONE <c-string 1..255>	INF=*ALL
Login-Verzeichnis	var(*LIST).DIR	S	<posix-pathname 1..1023>	INF=*ALL
POSIX-Gruppennummer	var(*LIST).GROUP-NUM	I	<integer 0..60002>	INF=*ALL
Default-POSIX-Gruppennummer	var(*LIST).GROUP-NUM-DEF	B	FALSE TRUE	INF=*ALL
Name des Programms, anhand dessen die BS2000-Benutzerkennung ausgewählt wird	var(*LIST).PROG	S	*SHELL <posix-pathname 1..1023>	INF=*ALL
Katalogkennung des Pubsets	var(*LIST).PUBSET	S	<catid 1..4>	INF=*ALL/ *USER-LIST

Fortsetzung ➔

Ausgabe-Information	Name der S-Variablen	T	Inhalt	Bedingung
BS2000-Benutzerkennung, deren POSIX-Benutzerattribute angezeigt werden	var(*LIST).USER-ID	S	<name 1..8>	INF=*ALL
	var(*LIST).USER-ID(*LIST)	S	<name 1..8>	INF=*USER-LIST
POSIX-Benutzernummer	var(*LIST).USER-NUM	I	<integer 0..60002>	INF=*ALL
Default-POSIX-Benutzernummer	var(*LIST).USER-NUM-DEF	B	FALSE TRUE	INF=*ALL

Nähere Informationen zu S-Variablen finden Sie im Handbuch „Kommandos Band 4“ [27].

## Beispiel 1

```
/declare-var var-name=pos-user-att(type=*struct),multi-elem=*list
/exec-cmd (show-posix-user-attr inf=*all),text-output=*none,structure-output=pos-user-att
/show-var pos-user-att,inf=*par(value=*c-literal)

POS-USER-ATT(*LIST).PUBSET = '1SBZ'
POS-USER-ATT(*LIST).USER-ID = 'TSOS'
POS-USER-ATT(*LIST).USER-NUM = 0
POS-USER-ATT(*LIST).USER-NUM-DEF = FALSE
POS-USER-ATT(*LIST).GROUP-NUM = 0
POS-USER-ATT(*LIST).GROUP-NUM-DEF = FALSE
POS-USER-ATT(*LIST).COMMENT = '*NONE'
POS-USER-ATT(*LIST).DIR = '/'
POS-USER-ATT(*LIST).PROG = '*SHELL'
*END-OF-VAR
/
```

## Beispiel 2

```
/exec-cmd (show-posix-user-attr inf=*user-list),text-outp=*none,struct-outp=pos-user-att
/show-var pos-user-att,inf=*par(value=*c-literal)

POS-USER-ATT(*LIST).PUBSET = '1SBZ'
POS-USER-ATT(*LIST).USER-ID(*LIST) = 'TSOS'
*END-OF-VAR
/
```

## SHOW-POSIX-USER-DEFAULTS

### Standardwerte für POSIX-Benutzerattribute anzeigen

**Anwendungsbereich:** USER-ADMINISTRATION

**Privilegierung:**      POSIX-ADMINISTRATION,  
                                  USER-ADMINISTRATION,  
                                  STD-PROCESSING

Dieses Kommando zeigt die POSIX-Standardattribute im Benutzerkatalog des angegebenen Pubsets an. Folgende Benutzer dürfen es ausführen:

- Inhaber des Privilegs POSIX-ADMINISTRATION oder USER-ADMINISTRATION für alle Pubsets.
- Gruppenverwalter der Gruppe \*UNIVERSAL auf dem von ihnen verwalteten Pubset.

#### SHOW-POSIX-USER-DEFAULTS

**PUBSET** = \*HOME / \*ALL / list-poss(20): <catid 1..4>

,**OUTPUT** = list-poss(2): \*SYSOUT / \*SYSLST(...)

\*SYSLST(...)

    | **SYSLST-NUMBER** = \*STD / <integer 1..99>

    | **,LINES-PER-PAGE** =64 / <integer 20..255>

#### **PUBSET =**

Pubset, aus dessen Benutzerkatalog die POSIX-Standardattribute angezeigt werden sollen.

#### **PUBSET = \*HOME**

Die POSIX-Standardattribute werden aus dem Benutzerkatalog des Home-Pubsets angezeigt.

#### **PUBSET = \*ALL**

Die POSIX-Standardattribute werden aus den Benutzerkatalogen aller Pubsets, die zum Zeitpunkt der Kommandoeingabe verfügbar sind, angezeigt.

#### **PUBSET = list-poss(20): <catid 1..4>**

Die POSIX-Standardattribute werden aus dem Benutzerkatalog des angegebenen Pubsets angezeigt.

**OUTPUT =**

Systemdatei für die Ausgabe der Information.

**OUTPUT = \*SYSOUT**

Die Information wird in die Systemdatei SYSOUT ausgegeben.

**OUTPUT = \*SYSLST(...)**

Die Information wird in die Systemdatei SYSLST ausgegeben.

**SYSLST-NUMBER =**

Bestimmt die SYSLST-Nummer.

**SYSLST-NUMBER = \*STD**

Bestimmt die Standard-SYSLST-Ausgabe.

**SYSLST-NUMBER = <integer 1..99>**

Bestimmt die angegebene SYSLST-Nummer.

**LINES-PER-PAGE =**

Gibt die Zeilenzahl pro Seite an.

**LINES-PER-PAGE = 64**

Standardmäßig werden 64 Zeilen pro Seite gedruckt.

**LINES-PER-PAGE = <integer 20..255>**

Die angegebene Zeilenzahl wird pro Seite gedruckt.

**Kommando-Returncodes**

<b>(SC2)</b>	<b>SC1</b>	<b>Maincode</b>	<b>Bedeutung</b>
2	0	CMD0001	Kommando fehlerfrei ausgeführt
	0	SRM6001	Kommando mit Warnung ausgeführt
	32	CMD2009	Fehler beim Erzeugen der Ausgabe-Variablen
	32	SRM6020	Kommando wegen eines Systemfehlers abgewiesen
	64	OPS0002	K2-Unterbrechung bei Ausgabe in S-Variable
	64	SRM6040	Kommando mit Fehlermeldung abgewiesen
	130	OPS0001	Ressourcenmangel bei Ausgabe in S-Variable
	130	SRM6030	Kommando wegen Ressourcenmangel abgewiesen

*Beispiel*

```
/SHOW-POSIX-USER-DEFAULTS PUBSET=A
```

*Ausgabe:*

```
/SHOW-POSIX-USER-DEFAULTS PUBSET=A
POSIX-USER-DEFAULTS --- PUBSET A
```

```
1998-02-18 14:14:05
```

```
-----  
USER-NUMBER      200  
GROUP-NUMBER     8  
COMMENT          POSIX public userID  
DIRECTORY        /home/usr0/gast  
PROGRAM          *SHELL
```

```
-----  
POSIX-USER-DEFAULTS
```

```
END OF DISPLAY
```

**S-Variablen**

Ausgabe-Information	Name der S-Variablen	T	Inhalt	Bedingung
Kommentar	var(*LIST).COMMENT	S	*NONE <c-string 1..255>	
Login-Verzeichnis	var(*LIST).DIR	S	<posix-pathname 1..1023>	
POSIX-Gruppennummer	var(*LIST).GROUP-NUM	I	<integer 0..60002>	
Name des Programms	var(*LIST).PROG	S	*SHELL <posix-pathname 1..1023>	
Katalogkennung des Pubsets	var(*LIST).PUBSET	S	<catid 1..4>	
POSIX-Benutzernummer	var(*LIST).USER-NUM	I	<integer 0..60002>	

Nähere Informationen zu S-Variablen finden Sie im Handbuch „Kommandos Band 4“ [27].

**Beispiel**

```
/declare-var var-name=POS-USER-DEF(type=*struct),multi-elem=*list
/exec-cmd cmd=(show-posix-user-defaults pubset=a), -
/           structure-output=POS-USER-DEF,text-output=*none
/show-var var-name=POS-USER-DEF
POS-USER-DEF(*LIST).PUBSET = A
POS-USER-DEF(*LIST).USER-NUM = 100
POS-USER-DEF(*LIST).GROUP-NUM = 1
POS-USER-DEF(*LIST).COMMENT = 'Systemadministration'
POS-USER-DEF(*LIST).DIR = /home/bs2000
POS-USER-DEF(*LIST).PROG = *SHELL
```

## SHOW-USER-ATTRIBUTES

### Informationen über die Einträge im Benutzerkatalog ausgeben

**Anwendungsbereich:** ACCOUNTING,  
USER-ADMINISTRATION

**Privilegierung:** USER-ADMINISTRATION, SECURITY-ADMINISTRATION,  
SAT-FILE-MANAGEMENT, SAT-FILE-EVALUATION,  
STD-PROCESSING, HARDWARE-MAINTENANCE

Dieses Kommando zeigt die Attribute einer BS2000-Benutzerkennung an, die mit den Kommandos /ADD-USER und /MODIFY-USER-ATTRIBUTES vereinbart wurden.

Die POSIX-Benutzerattribute können gesondert durch das Kommando /SHOW-POSIX-USER-ATTRIBUTES angezeigt werden (siehe Seite 145).

Folgende Benutzer dürfen das Kommando /SHOW-USER-ATTRIBUTES ausführen:

- Inhaber des Privilegs USER-ADMINISTRATION für alle BS2000-Benutzerkennungen.
- Gruppenverwalter für die BS2000-Benutzerkennungen, die ihren Gruppen zu- und untergeordnet sind.
- jeder Benutzer für seine eigene BS2000-Benutzerkennung.

An der Benutzeroberfläche des Kommandos /SHOW-USER-ATTRIBUTES ändert sich durch POSIX nichts. Das Kommando ist in „Kommandos Band 3“ [27] beschrieben.



Das Kommando /SHOW-USER-ATTRIBUTES zeigt die POSIX-Abrechnungsnummer an, falls sie festgelegt wurde. Andernfalls wird \*NONE ausgegeben.

Die POSIX-Abrechnungsnummer wird für Gruppen in den Gruppenkommandos nicht berücksichtigt.

## S-Variablen

Die Information zur Abrechnungsnummer für *rlogin* wird in folgender S-Variablen abgelegt:

Ausgabe-Information	Name der S-Variablen	T	Inhalt	Bedingung
Abrechnungsnummer beim POSIX-Zugang über rlogin	var(*LIST).ACCOUNT(*LIST).POSIX-RLOG-DEF	S	*NO *YES	INF=*ATTR

Nähere Informationen zu S-Variablen finden Sie im Handbuch „Kommandos Band 4“ [27].

## START-POSIX-SHELL

### POSIX-Shell zur Verfügung stellen

**Anwendungsbereich:** PROCEDURE, UTILITIES

**Privilegierung:** STD-PROCESSING

Dieses Kommando startet die POSIX-Shell. Nach erfolgreichem Zugang zur POSIX-Shell kann der Benutzer POSIX-Kommandos eingeben (siehe Abschnitt „Kommando umfang der Basis-Shell“ auf Seite 179) und Handbuch „POSIX-Kommandos“ [2]).

Beendet wird die POSIX-Shell mit dem POSIX-Kommando *exit*.

START-POSIX-SHELL	Kurzname: <b>POSIX-SHELL</b>
<b>VERSION</b> = <u>*STD</u> / <product-version without-man-corr>	
<b>,MONJV</b> = <u>*NONE</u> / <filename 1..54 without-gen-vers>	
<b>,CPU-LIMIT</b> = <u>*JOB-REST</u> / <integer 1..32767>	

**VERSION** = \*STD / <product-version without-man-corr>

Versionsnummer des aufzurufenden Programms (hier der POSIX-Shell).

Voreingestellt ist \*STD, d.h. es wird die aktuell verfügbare Version aufgerufen.

**MONJV** = \*NONE / <filename 1..54 without-gen-vers>

Name der Jobvariablen, die das Programm überwachen soll. Wenn die Jobvariable noch nicht vorhanden ist, wird sie neu angelegt. Während des Programmablaufs setzt das System die Jobvariable auf die entsprechenden Werte:

\$R Programm läuft

\$T Programm beendet

\$A Programm abnormal beendet

Dieser Operand steht nur zur Verfügung, wenn das Softwareprodukt „Jobvariablen“ im Einsatz ist (siehe auch Handbuch „Jobvariablen“ [24]).

**CPU-LIMIT = \*JOB-REST / <integer 1..32767 seconds>**

Maximale CPU-Zeit in Sekunden, die das Programm beim Ablauf benötigen darf. Wenn der Auftrag ohne Zeitbeschränkung (CPU-LIMIT=\*NO) gestartet wurde, läuft das Programm ebenfalls ohne Zeitbeschränkung, d.h. die Angabe wird ignoriert.

Wenn das Programm beim Ablauf die angegebene Zeit überschreitet, wird im Dialog das Programm unterbrochen und eine Meldung ausgegeben. Der Benutzer kann dann einen Dump anfordern, das Programm abbrechen oder forsetzen. Wenn im Programm eine STXIT-Routine für das Erreichen des CPU-Limits definiert ist, wird diese durchlaufen und das Programm beendet.

Im Stapelbetrieb wird das Programm bei Zeitüberschreitung beendet.

**CPU-LIMIT = \*JOB-REST**

Wenn der Auftrag mit Zeitbeschränkung gestartet wurde, wird für das Programm der Wert verwendet, der bei der Systemgenerierung festgelegt wurde. Andernfalls läuft das Programm ohne Zeitbeschränkung.

**Kommando-Returncodes**

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler

*Beispiel*

Siehe „Beispielsitzung“ auf Seite 56.

## 9 POSIX-Meldungen

Das Subsystem POSIX gibt seine Meldungen über das im BS2000 übliche Meldungsverfahren auf die BS2000-Konsole aus.

Die POSIX-Meldungsnummern sind in folgende Nummernkreise unterteilt.

Nummernkreis	POSIX-Komponente
POS1000 - POS1100	Portierter Systemkern
POS2000 - POS2999	POSIX-Initialisierung
POS3000 - POS3999	POSIX-Terminierung
POS4000 - POS4999	POSIX-Systemprozesse und deren Fehlerkorrektur (error recovery)
POS5000 - POS5999	Fehlerkorrektur allgemein

Die Meldungen aus dem portierten Code haben folgendes Format:

POS1xxx Meldung des POSIX-Kernels: (&00).

POS1xxx Message of the POSIX Kernel: (&00).

(&00) beginnt mit der Fehlerstufe (z.B. `Warning:`) und enthält den ursprünglichen Meldungstext des portierten Codes.

Zu den portierten Meldungen ist keine Ausgabe von erweiterten Informationen möglich.

POSCOPY Copyright (C) '(&00)' '(&01)' All Rights Reserved  
POSCOPY Copyright (C) '(&00)' '(&01)' Alle Rechte vorbehalten  
POSLOAD Program '(&00)', Version '(&01)' of '(&02)' loaded from file '(&03)'  
POSLOAD Programm '(&00)', Version '(&01)' vom '(&02)' aus Datei '(&03)' geladen  
POSSTRT Procedure '(&00)', Version '(&01)' of '(&02)' started from file '(&03)'  
POSSTRT Prozedur '(&00)', Version '(&01)' vom '(&02)' aus Datei '(&03)' gestartet  
POS1000 (&00)  
POS1000 (&00)  
(B) Routing code: # Weight: 30

**Bedeutung**

Statusmeldung des POSIX-Subsystems.

POS1010 Message of the POSIX kernel: (&00)  
POS1010 Meldung des POSIX-Kernels: (&00)  
(B) Routing code: # Weight: 50

**Bedeutung**

Hinweis des POSIX-Subsystems.

POS1020 Message of the POSIX kernel:(&00)  
POS1020 Meldung des POSIX-Kernels: (&00)  
(B) Routing code: # Weight: 70

**Bedeutung**

Warnung des POSIX-Subsystems.

**Maßnahme**

Den Systemverwalter verstaendigen.

POS1030 Message of the POSIX kernel: (&00)  
POS1030 Meldung des POSIX-Kernels: (&00)  
(B) Routing code: # Weight: 99

**Bedeutung**

Nicht behebbarer Fehler im POSIX-Subsystem.

Die Terminierung des POSIX-Subsystems wird eingeleitet.

**Maßnahme**

Den Systemverwalter verstaendigen.

POS2000 Subsystem private initialization of POSIX has been processed  
POS2000 Subsystem-private Initialisierung von POSIX abgeschlossen  
(C) Routing code: # Weight: 99

**Bedeutung**

Die subsystem-private Initialisierung von POSIX wurde erfolgreich durchgefuehrt  
(Hinweis: POSIX ist erst nutzbar, wenn DSSM-Initialisierung (Meldung ESM0220) und  
Initialisierungsaktionen des Init-Prozesses (Meldung POS4100) abgeschlossen sind.)

- POS2001 Subsystem private initialisation of POSIX has been proceeded  
(on base of the generic root file system)
- POS2001 Subsystem-private Initialisierung des POSIX-Subsystems abgeschlossen  
(Start auf generischem Root-Dateisystem)  
(B) Routing code: # Weight: 99
- Bedeutung**  
Start auf generischem Root-Dateisystem ist nur während der POSIX-Erstinstallation zugelassen; in diesem Fall wird kein init-Prozess gestartet, sondern die Erstinstallation wird fortgesetzt.
- Maßnahme**  
1.) Erstinstallation: keine.  
2.) Versehentlicher Start auf dem generischen Root-Dateisystem:  
POSIX terminieren, anschliessend den Namen des gewünschten Root-Dateisystems in die POSIX-Informationsdatei eintragen und POSIX neu starten.
- POS2010 Error occurred on reading POSIX information file  
Fehler beim Lesen der POSIX-Informationsdatei
- POS2010 (B) Routing code: # Weight: 99
- Bedeutung**  
POSIX-Start wird mit den Standardwerten fortgesetzt.
- POS2011 No reservation of system memory possible (class 4)  
Fehler beim Reservieren von Speicher (Speicherklasse 4)
- POS2011 (B) Routing code: # Weight: 99
- Bedeutung**  
Die POSIX-Informationsdatei kann nicht eröffnet werden weil kein Speicher reserviert werden kann. POSIX-Start wird mit den Standardwerten fortgesetzt.
- POS2012 The POSIX information file could not be opened  
Kein Öffnen der POSIX-Informationsdatei möglich
- POS2012 (B) Routing code: # Weight: 99
- Bedeutung**  
Der POSIX-Start wird fortgesetzt. An Stelle der Parameter aus der POSIX-Informationsdatei werden die entsprechenden Standardwerte benutzt.
- POS2013 Record pointer could not be positioned in POSIX information file  
Fehler beim Positionieren des Satzzeigers in der POSIX-Informationsdatei
- POS2013 (B) Routing code: # Weight: 99
- Bedeutung**  
Die POSIX-Informationsdatei wird geschlossen; der POSIX-Start wird mit den Standardwerten fortgesetzt.

POS2014 Error occurred during CLOSE of POSIX information file  
POS2014 Fehler beim Schliessen der POSIX-Informationsdatei  
(B) Routing code: # Weight: 99

**Maßnahme**

Bitte den Systemverwalter verstaendigen.

POS2015 POSIX cannot be started (REBOOT of first installation)  
POS2015 POSIX kann nicht gestartet werden (REBOOT der Erstinstallation)  
(B) Routing code: # Weight: 99

**Bedeutung**

Die Probleme, die beim POSIX-Start aufgetreten sind, sind durch entsprechende Konsol-  
meldungen im Detail beschrieben.

POS2016 Error occurred during erase of generic root file system (userid 'SYSROOT')  
POS2016 Fehler beim Loeschen des generischen Root-Dateisystems unter der  
Benutzerkennung 'SYSROOT'  
(B) Routing code: # Weight: 99

**Maßnahme**

Bitte den Systemverwalter informieren.

POS2017 Userid 'SYSROOT' not available or locked  
POS2017 Benutzerkennung 'SYSROOT' nicht verfuegbar oder gesperrt  
(C) Routing code: # Weight: 99

**Maßnahme**

/UNLOCK-USER USER-IDENTIFICATION=SYSROOT zum Entsperren der Benutzer-  
kennung 'SYSROOT' eingeben.

POS2018 No information available on userid 'SYSROOT'  
POS2018 Keine Information verfuegbar ueber die Benutzerkennung 'SYSROOT'  
(C) Routing code: # Weight: 99

**Bedeutung**

Bitte die Details der SERSLOG-Datei entnehmen (\$\$JINFO-Returncode).

POS2019 Error in call of IMON interface '(&00)' for the enter file of the init process  
POS2019 Fehler beim Aufruf der IMON-Schnittstelle '(&00)' fuer die Enter-Datei des Init-  
Prozesses  
(C) Routing code: # Weight: 99

**Bedeutung**

Der Dateiname der Enter-Datei fuer den Init-Prozess kann nicht ermittelt werden.

**Maßnahme**

Bitte den Systemverwalter informieren.

POS2020 Error while constructing the filename for the enter file of the init process  
POS2020 Fehler beim Konstruieren des Dateinamens fuer die Enter-Datei des Init-Prozesses  
(C) Routing code: # Weight: 99

**Bedeutung**

Der Dateiname der Enter-Datei fuer den Init-Prozess kann nicht ermittelt werden.

**Maßnahme**

Bitte den Systemverwalter informieren.

POS2100 Parameter file for automatic POSIX installation does not exist  
POS2100 Parameterdatei fuer POSIX-Installation im automatischen Betrieb existiert nicht

**Maßnahme**

Bitte Parameterdatei fuer den automatisierten Betrieb bereitstellen.

POS2200 Signal mask of process with pid (&00) destroyed. No further information available

POS2200 Signalmaske des Prozesses mit pid (&00) zerstoert; keine weiteren Informationen ueber den Prozess verfuegbar

**Maßnahme**

Bitte Systemverwaltung informieren (mit BS2000-Diagnose-Tools kann weitere Information ueber den fehlerhaften Prozess gewonnen werden).

POS2201 Process with pid (&00) is a zombie. No further information available  
POS2201 Prozess mit pid (&00) befindet sich im Zombie-Zustand; keine weiteren Informationen verfuegbar

POS2202 Process structur of process with pid (&00) already released. No further information available

POS2202 Prozess-Struktur des Prozesses mit der pid (&00) bereits freigegeben. Keine weiteren Informationen verfuegbar

POS2203 Task with TSN (&00) has no process structure. No further information available  
POS2203 Task mit TSN (&00) besitzt keine Prozessstruktur. Keine weiteren Informationen verfuegbar

**Maßnahme**

Bitte den Systemverwalter verstaendigen (mit BS2000-Diagnose-Tools kann eventuell weitere Information ueber die Task gewonnen werden).

POS2204 Task with TSN (&00) is no POSIX process. No further information available  
POS2204 Task mit TSN (&00) ist kein POSIX-Prozess. Keine weiteren Informationen verfuegbar

**Maßnahme**

Bitte Systemverwalter verstaendigen (mit BS2000-Diagnose-Tools kann eventuell mehr Information ueber die Task gewonnen werden).

- POS2800 Diagnostic tool of subsystem POSIX creates a system dump  
POS2800 POSIX-Diagnosetool erstellt den angeforderten Systemdump  
**(B) Routing code: # Weight: 99**
- POS2900 Procedure POSADDUS: The account number for rlogin could not be inserted into the user data  
POS2900 Prozedur POSADDUS: Die Accountnummer fuer rlogin konnte nicht eingetragen werden
- POS2901 Procedure POSADDUS: The user 'TSOS' has not been privileged as root  
POS2901 Prozedur POSADDUS: Der Benutzer 'TSOS' hat nicht die Root-Berechtigung erhalten
- POS2902 Procedure POSADDUS: The userid does not exist  
POS2902 Prozedur POSADDUS: Die Benutzerkennung existiert nicht
- POS2903 Procedure POSADDUS: The home directory could not be inserted into the POSIX user data  
POS2903 Prozedur POSADDUS: Der Name des Home-Verzeichnisses konnte nicht in die POSIX-Benutzerdaten eingetragen werden
- POS2904 Procedure POSADDUS: The group number could not be inserted into the POSIX user data  
POS2904 Prozedur POSADDUS: Die Gruppennummer konnte nicht in die POSIX-Benutzerdaten eingetragen werden
- POS2905 Procedure POSADDUS: The program name could not be inserted into the POSIX user data  
POS2905 Prozedur POSADDUS: Der Programmname konnte nicht in die POSIX-Benutzerdaten eingetragen werden
- POS2906 Procedure POSADDUS has been processed without error  
POS2906 Prozedur POSADDUS fehlerfrei abgelaufen
- POS3000 Error during check of tasks connected to subsystem POSIX  
POS3000 Fehler bei Ueberpruefung von Tasks, die an das POSIX-Subsystem konnektiert sind  
**(B) Routing code: # Weight: 99**

### **Bedeutung**

Fehler bei der Ueberpruefung, ob sich eine konnektierte Task bereits in der Phase der Programm- oder Taskterminierung befindet (mit der Schnittstelle \$EZRTSW).  
Fehler-Code in der SERSLOG-Datei abgelegt.

- POS3001 Subsystem POSIX has not been locked against new users  
POS3001 Das POSIX-Subsystem wurde nicht fuer neue Benutzer gesperrt  
**(B) Routing code: # Weight: 99**

### **Bedeutung**

Zu Beginn der regulaeren Terminierung sperrt der Init-Prozess das POSIX-Subsystem fuer neue Benutzer; bei Fehler sperrt DSSM das Subsystem und es wird eine abnormale Subsystemterminierung eingeleitet.

### **Maßnahme**

Bitte den Systemverwalter informieren.

POS3002 The command /HOLD-SUBSYSTEM POSIX is not allowed  
POS3002 Das Kommando /HOLD-SUBSYSTEM POSIX ist nicht zugelassen  
(B) Routing code: # Weight: 99

**Bedeutung**

Das Kommando wurde vom POSIX-Subsystem zurueckgewiesen. Das POSIX-Subsystem laeuft unveraendert weiter (neue Anwendungen koennen gestartet werden.)

POS3003 POSIX is terminated abnormally, because init process does not exist  
POS3003 POSIX wird abnormal terminiert, weil kein Init-Prozess existiert  
(C) Routing code: # Weight: 99

**Bedeutung**

Waehrend der regulaeren Subsystemterminierung wurde festgestellt, dass der Init-Prozess bereits terminiert ist; deshalb wird POSIX abnormal beendet.

POS3004 Error during check of connected tasks  
POS3004 Fehler bei Ueberpruefung konnektierter Tasks  
(B) Routing code: # Weight: 99

**Bedeutung**

Die Subsystem-Informationsfunktion \$ESMSWS meldet einen Fehler (waehrend der Subsystemterminierung). Der Fehlercode ist in der SERSLOG-Datei abgelegt. Falls moeglich, wird die POSIX-Terminierung im abnormalen Modus fortgesetzt.

POS3005 No check of tasks connected to POSIX possible due to memory shortage  
POS3005 Wegen Speichermaengels keine Ueberpruefung von konnektierten Tasks moeglich  
(B) Routing code: # Weight: 99

**Bedeutung**

Die Informationsfunktion \$ESMSWS konnte nicht aufgerufen werden weil der noetige Ausgabebereich nicht angelegt werden konnte. Das POSIX-Subsystem wird automatisch abnormal terminiert, soweit moeglich.

POS3007 Wait on init process interrupted. POSIX termination continues in abnormal mode  
POS3007 Warten auf POSIX-Terminierung durch Init-Prozess abgebrochen. Die POSIX-Terminierung wird im abnormalen Modus fortgesetzt  
(B) Routing code: # Weight: 99

POS3009 Regular termination of POSIX interrupted because the subsystem is still in use.  
POS3009 New connections are not possible

POS3009 Regulaere POSIX-Terminierung abgebrochen, weil einige POSIX-Anwendungen noch nicht beendet sind (keine Neukonnektierung an das POSIX-Subsystem moeglich)  
(B) Routing code: # Weight: 99

**Bedeutung**

Die Ueberpruefung von Tasks, die noch an das Subsystem konnektiert sind (in der subsystemprivaten CLOSE-CONTROL-Routine), war positiv; die regulaere POSIX-Terminierung wurde deshalb abgebrochen.

Tasks, die bereits konnektiert sind, koennen das Subsystem weiterhin benutzen.

**Maßnahme**

Keine Massnahme durch den Subsystemverwalter notwendig. Das Subsystem kann bei Bedarf forciert terminiert werden (/STOP-SUBSYSTEM-Kommando mit Parameter SUBSYSTEM-PARAMETER='FORCED-BY-SUBSYSTEM').

POS3010 Subsystem POSIX has been terminated

POS3010 Das POSIX-Subsystem wurde beendet

(B) Routing code: # Weight: 99

POS3011 Normal POSIX termination during BS2000 shutdown

POS3011 Normale POSIX-Terminierung waehrend BS2000 Shutdown

(B) Routing code: # Weight: 99

**Bedeutung**

Der Init-Prozess wurde waehrend des Shutdown terminiert; anschliessend wird POSIX forciert beendet.

POS3012 Error occurred during POSIX termination (reboot of initial installation)

POS3012 Fehler bei POSIX-Terminierung waehrend Reboot (Erstinstallation)

(B) Routing code: # Weight: 99

**Bedeutung**

Reboot wird abgebrochen.

POS3013 Error occurred on erase of generic root file system (copy under userid 'SYSROOT') during initial installation

POS3013 Fehler beim Loeschen der Kopie des generischen Rootdateisystems unter der Benutzerkennung 'SYSROOT' (Erstinstallation)

(B) Routing code: # Weight: 99

**Maßnahme**

Den Systemverwalter informieren.

POS3015 Subsystem private STOP-COMMISSION routine has been processed (POSIX termination during BS2000 shutdown)

POS3015 Subsystem-private STOP-COMMISSION-Routine beendet (POSIX-Terminierung waehrend BS2000 Shutdown)

(B) Routing code: # Weight: 99

POS3016 Error during check of subsystem status of POSIX

POS3016 Fehler beim Ueberpruefen des Subsystemstatus von POSIX

(B) Routing code: # Weight: 99

POS3018 Wait for POSIX termination interrupted. Shutdown continues

POS3018 Warten auf POSIX-Terminierung abgebrochen; Shutdown wird fortgesetzt

(B) Routing code: # Weight: 99

POS3019 Wait for forced POSIX termination interrupted; shutdown continues

POS3019 Warten auf forciertere POSIX-Terminierung abgebrochen. Shutdown wird fortgesetzt

(B) Routing code: # Weight: 99

POS3020 New processes are created by fork() in spite of POSIX termination  
POS3020 Die Erzeugung neuer POSIX-Prozesse durch fork() ist noch nicht abgeschlossen  
(POSIX-Terminierung)  
(B) Routing code: # Weight: 99

**Bedeutung**

Das Warten auf die Verarbeitung von fork-Aufrufen wurde abgebrochen.  
Die POSIX-Terminierung wird im abnormalen Modus weitergefuehrt.

POS3021 Wait on new processes created by fork has been interrupted (during abnormal  
POSIX termination)

POS3021 Warten auf Verarbeitung von fork-Prozessen abgebrochen (waehrend abnormaler  
POSIX-Terminierung)  
(B) Routing code: # Weight: 99

**Bedeutung**

Das Warten auf Prozesse, die sich gerade im Stadium der Erzeugung durch fork() befinden,  
wird abgebrochen.

POS3022 STOP-COMMISSION routine processed (forced POSIX termination during BS2000  
shutdown)

POS3022 STOP-COMMISSION-Routine beendet (forcierte POSIX-Terminierung waehrend BS2000  
Shutdown)

(B) Routing code: # Weight: 99

POS3023 Wait for init process. POSIX termination continues

POS3023 Warten auf Terminierungsaktionen des Init-Prozesses. POSIX-Terminierung wird  
fortgesetzt

(B) Routing code: # Weight: 99

POS3024 SPOOL server has not been disconnected from POSIX subsystem during POSIX  
termination

POS3024 SPOOL Server diskonnektiert sich nicht vom POSIX-Subsystem waehrend der POSIX-  
Terminierung

(B) Routing code: # Weight: 99

**Maßnahme**

Bitte den Systemverwalter informieren (SLED als Diagnoseunterlage noetig).

POS3025 At first call of command /STOP-SUBSYSTEM the operand SUBSYSTEM-PARAMETER= must  
not be used

POS3025 SUBSYSTEM-PARAMETER= beim ersten Aufruf des Kommandos /STOP-SUBSYSTEM nicht  
zugelassen

(B) Routing code: # Weight: 99

**Bedeutung**

Der Operand SUBSYSTEM-PARAMETER= ist fuer den Fall der forcierten Terminierung  
reserviert.

POS3026 Wrong value of operand SUBSYSTEM-PARAMETER=  
POS3026 Fehler im Operanden SUBSYSTEM-PARAMETER=  
(B) Routing code: # Weight: 99

**Maßnahme**

Das Kommando /STOP-SUBSYSTEM SUBSYSTEM-NAME=POSIX, SUBSYSTEM-PARAMETER='FORCED-BY-SUBSYSTEM' benutzen.

POS3027 No operand SUBSYSTEM-PARAMETER= given in second call of command /STOP-SUBSYSTEM SUBSYSTEM-NAME=POSIX

POS3027 Angabe des Operanden SUBSYSTEM-PARAMETER= fehlt beim zweiten Aufruf des Kommandos /STOP-SUBSYSTEM SUBSYSTEM-NAME=POSIX  
(B) Routing code: # Weight: 99

POS3800 No disconnect of task (&00) from entry POSCALL because of an error  
POS3800 Fehler beim Diskonnektieren der Task (&00) vom Entry POSCALL  
(B) Routing code: # Weight: 99

**Bedeutung**

Der Fehlercode der Schnittstelle \$ESMDCN ist in der SERSLOG-Datei eingetragen.

**Maßnahme**

Bitte den Systemverwalter benachrichtigen.

POS3801 Error occurred during disconnection from POSIX subsystem  
POS3801 Fehler beim Diskonnektieren vom POSIX-Subsystem  
(B) Routing code: # Weight: 99

**Bedeutung**

Fehler beim Aufruf der Schnittstelle \$ESMDCN (RETURN=DSL).  
Der Fehlercode ist in der SERSLOG-Datei eingetragen.

POS3900 Termination of connected tasks during subsystem termination of POSIX  
POS3900 POSIX-Terminierung fuehrt zur Terminierung noch konnektierter Tasks  
(B) Routing code: # Weight: 99

**Bedeutung**

Taskterminierung durch subsystemprivate STOP-COMMISSION-Routine eingeleitet.

POS3901 Abnormal task termination because of an error during subsystem termination  
POS3901 Abnormale Taskterminierung wegen eines Fehlers in der POSIX-Terminierung  
(B) Routing code: # Weight: 99

**Bedeutung**

Taskterminierung durch subsystemprivate CLOSE-CONTROL-Routine.

POS3902 Task has been terminated normally in order to disconnect from the subsystem POSIX  
POS3902 Die Task wurde normal beendet, um sie vom POSIX-Subsystem zu diskonnektieren  
(B) Routing code: # Weight: 99

- POS3903 Abnormal task termination because of error during check of the task state during subsystem termination  
POS3903 Abnormale Taskterminierung wegen Fehler beim Check des Taskzustandes waehrend der Subsystemterminierung  
(B) Routing code: # Weight: 99

**Bedeutung**

Taskterminierung durch subsystemprivate STOP-COMMISSION Routine.

- POS3904 Abnormal task termination after panic()  
POS3904 Abnormale Taskterminierung, nachdem die Task auf einen panic() gelaufen ist  
(B) Routing code: # Weight: 99

- POS3905 POSIX system server has terminated because of an error  
POS3905 POSIX-Systemprozess wegen eines Fehlers terminiert  
(B) Routing code: # Weight: 99

- POS3906 Abnormal task termination because of POSIX termination (task had been created by fork())  
POS3906 Die Terminierung des POSIX-Subsystems hat eine abnormale Taskterminierung eingeleitet (Task durch fork() erzeugt)  
(B) Routing code: # Weight: 99

- POS3907 Rlogin rejected by system exit 030  
POS3907 Rlogin abgewiesen durch System-Exit 030  
(B) Routing code: # Weight: 99

- POS4000 Termination of subsystem POSIX because of termination of a POSIX system process  
POS4000 POSIX-Terminierung wegen Absturz eines POSIX-Systemprozesses  
(C) Routing code: # Weight: 99

- POS4010 BS2000 file '(&00)' not found  
POS4010 BS2000-Datei '(&00)' nicht gefunden  
(C) Routing code: # Weight: 99

**Bedeutung**

FSTAT fuer die BS2000-Datei wurde fehlerhaft ausgefuehrt.

**Maßnahme**

FSTAT ausfuehren und den Fehler beseitigen.

- POS4011 Error in READ/WRITE server  
POS4011 Fehler im READ/WRITE-Server  
(C) Routing code: # Weight: 99

**Bedeutung**

Ein BS2000-Makro wurde fehlerhaft ausgefuehrt.  
Der Read/Write-Server wurde abnormal terminiert.

**Maßnahme**

Systemverwalter verstaendigen. Der Fehler wurde in der SERSLOG-Datei protokolliert.

POS4012 Error in OPEN/CLOSE server  
POS4012 Fehler im OPEN/CLOSE-Server  
(C) Routing code: # Weight: 99

**Bedeutung**

Ein BS2000-Makro wurde fehlerhaft ausgefuehrt.  
Der Open/Close-Server wurde abnormal terminiert.

**Maßnahme**

Den Systemverwalter verstaendigen. Der Fehler wurde in der SERSLOG-Datei protokolliert.

POS4013 Specified user id '(&00)' does not exist in HOME pubset  
POS4013 Die angegebene Benutzerkennung '(&00)' existiert nicht im HOME-Pubset  
(C) Routing code: # Weight: 99

**Bedeutung**

Es konnte keine Standard-Pubset-ID ermittelt werden.

**Maßnahme**

Die Benutzerkennung vom Systemverwalter im System eintragen lassen.  
Das Kommando 'mount' wiederholen.

POS4014 Filesystem '(&00)' could not be mounted due to protection of its container file  
POS4014 Dateisystem '(&00)' konnte nicht montiert werden weil seine Behaelterdatei  
geschuetzt ist  
(C) Routing code: # Weight: 99

POS4015 Disk with container file of file system '(&00)' not available  
POS4015 Platte mit der Behaelterdatei des Dateisystems '(&00)' nicht erreichbar  
(C) Routing code: # Weight: 99

**Bedeutung**

Die Platte mit dem montierten Dateisystem wurde waehrend des  
POSIX-Laufes weggenommen (z.B. /EXPORT-PUBSET, /DETACH-DEVICE).

**Maßnahme**

Um mit dem Dateisystem arbeiten zu koennen ist das Plattenproblem zu beseitigen, POSIX  
zu beenden und neu zu starten.

POS4016 File system '(&00)' could not be mounted because its container file has been  
migrated  
POS4016 Dateisystem '(&00)' konnte nicht montiert werden weil seine Behaelterdatei  
migriert ist  
(C) Routing code: # Weight: 99

POS4017 File system '(&00)' could not be mounted because its container file has attribute BLKCTRL=\*PAMKEY  
POS4017 Dateisystem '(&00)' konnte nicht montiert werden weil seine Behaelterdatei das Attribut BLKCTRL=\*PAMKEY hat  
(C) Routing code: # Weight: 99

**Bedeutung**

Die Behaelterdateien von POSIX-Dateisystemen muessen BLKCTRL=\*NO haben.

**Maßnahme**

Die Behaelterdatei des Dateisystems mit PAMCONV umwandeln und das Kommando mount wiederholen.

POS4051 Error in SOCKET/XTI server  
POS4051 Fehler im SOCKET/XTI-Server  
(B) Routing code: # Weight: 99

**Bedeutung**

Ein BS2000-Makro wurde fehlerhaft ausgefuehrt.  
Der SOCKET/XTI-Server wurde abnormal terminiert.

**Maßnahme**

Bitte den Systemverwalter benachrichtigen.  
Der Fehler wurde in der SERSLOG Datei protokolliert.

POS4100 INIT: (&00)  
POS4100 INIT: (&00)  
(C) Routing code: # Weight: 99

**Bedeutung**

Meldung des Init-Prozesses.

POS5000 Error occurred during panic() processing (subsystem deletion could not be initiated)  
POS5000 Bei der Einleitung der Subsystemterminierung waehrend der panic-Verarbeitung ist ein Fehler aufgetreten  
(C) Routing code: # Weight: 99

**Bedeutung**

Aufruf an DSSM zur Subsystemterminierung liefert Fehlercode (Schnittstelle \$ESMDEL).  
Moegliche Ursachen:  
Subsystem wird bereits terminiert, Fehler beim Aufruf oder in der Verarbeitung des \$ESMDEL. Der Fehlercode wird in der SERSLOG-Datei protokolliert.

**Maßnahme**

Bitte den Systemverwalter verstaendigen.

POS5001 No rlogin possible because of overload of job class  
POS5001 Kein rlogin moeglich wegen Erreichen des Jobklassengrenzwerts  
(C) Routing code: # Weight: 99

POS5002	No rlogin possible due to memory shortage Kein rlogin moeglich wegen Speichermangels (C) Routing code: # Weight: 99
POS5003	Termination of holder task because of severe error during POSIX start Terminierung der Holdertask wegen schweren Fehlers beim POSIX-Start (C) Routing code: C Weight: 99
POS5004	Initialization of subsystem POSIX has been interrupted because of an error Abbruch der POSIX-Initialisierung wegen eines Fehlers (C) Routing code: # Weight: 99

**Bedeutung**

Der Absturz der Holdertask des POSIX-Subsystems hat einen (DSSM-)Restart des Subsystems eingeleitet, der aber das Subsystem nicht vollstaendig initialisiert. Aus diesem Grund kann der Init-Prozess nicht anlaufen und das POSIX-Subsystem kann nicht genutzt werden.

**Maßnahme**

Nach der Meldung ESM0220 das POSIX-Subsystem per Kommando /STOP-SUBSYSTEM (ohne Parametereingabe) beenden; anschliessend kann das POSIX-Subsystem mit /START-SUBSYSTEM (und eventuell einer neuen REP-Datei) wieder gestartet werden.

POS5005	Container file of root file system does not exist (name of container file: see POSIX information file)
POS5005	Behaelterdatei des Root-Dateisystems ist nicht vorhanden (Name der Behaelterdatei: siehe POSIX-Informationsdatei)
	(C) Routing code: # Weight: 99

**Bedeutung**

Keine.

**Maßnahme**

Bitte gueltigen Dateinamen eintragen.

POS5006	FSTAT error (container file of root file system)
POS5006	FSTAT-Makro, angewandt auf Behaelterdatei des Root-Dateisystems, meldet Fehler
	(C) Routing code: # Weight: 99

**Bedeutung**

Der Fehlercode des FSTAT-Makros ist in der SERSLOG-Datei eingetragen.

POS5007	The root file system is not located under userid 'SYSROOT' Das Root-Dateisystem ist nicht unter der Kennung 'SYSROOT' abgelegt (C) Routing code: # Weight: 99
---------	---

**Bedeutung**

Aus Sicherheitsgruenden muss das POSIX-Root-Dateisystem unter der Kennung SYSROOT angelegt werden.

- POS5008 No rlogin possible due to shortage of resources  
POS5008 Kein rlogin moeglich wegen Ressourcenengpass  
(C) Routing code: # Weight: 99
- POS5009 The name of the root file system in POSIX information file is invalid  
POS5009 Der Name des Root-Dateisystems in der POSIX-Informationsdatei ist ungultig  
(C) Routing code: # Weight: 99

**Maßnahme**

Bitte gueltigen Root-Dateisystem-Namen in die POSIX-Informationsdatei eintragen.

- POS5010 Connection error between UFS and READ/WRITE server  
POS5010 Verbindungsfehler zwischen UFS und READ/WRITE-Server  
(C) Routing code: # Weight: 99

**Bedeutung**

Ein BS2000-Makro wurde fehlerhaft ausgefuehrt.  
Das POSIX Subsystem wird abnormal terminiert.

**Maßnahme**

Den Systemverwalter verstaendigen. Der Fehler wurde in der SERSLOG-Datei protokolliert.

- POS5011 Connection error between UFS and OPEN/CLOSE-Server  
POS5011 Verbindungsfehler zwischen UFS und OPEN/CLOSE-Server  
(C) Routing code: # Weight: 99

**Bedeutung**

Ein BS2000-Makro wurde fehlerhaft ausgefuehrt.  
Das POSIX-Subsystem wird abnormal terminiert.

**Maßnahme**

Den Systemverwalter verstaendigen. Der Fehler wurde in der SERSLOG-Datei protokolliert.

- POS5100 Internal error during fork() processing of son  
POS5100 Interner Fehler in Sohntask beim POSIX FORK-SVC

**Bedeutung**

Naeheres zur Art des Fehlers siehe in der SERSLOG-Datei.

**Maßnahme**

Das den fork() nutzende Programm spaeter nochmals aufrufen  
(z.B. im Fall von Systemueberlastung).

POS5200 Task was terminated due to a signal  
POS5200 Die Task wurde wegen eines Signals terminiert

POS5201 Wait for a lock has been interrupted by termination of process  
POS5201 Das Warten des POSIX-Prozesses auf einen Lock wurde mit Task-Terminierung abgebrochen

**(B) Routing code: # Weight: 99**

**Bedeutung**

Während der POSIX-Terminierung werden Tasks, die vergeblich auf einen Lock warten, beendet.

POS9000 POSIX '(&00)' '(&01)' '(&02)' Copyright (C) Siemens Nixdorf Informationssysteme AG 1994 All rights reserved.  
POS9000 POSIX '(&00)' '(&01)' '(&02)' Copyright (C) Siemens Nixdorf Informationssysteme AG 1994 Alle Rechte vorbehalten.

**Bedeutung**

Startmeldung der POSIX Shell

POS9001 The POSIX subsystem is not created  
POS9001 Das POSIX-Subsystem ist nicht gestartet

**Maßnahme**

Den Systemverwalter verständigen.

POS9002 Start of /usr/bin/slogin failed  
POS9002 Start von /usr/bin/slogin fehlerhaft

**Bedeutung**

/usr/bin/slogin ist im POSIX-Root-Dateisystem fehlerhaft.

**Maßnahme**

Den Systemverwalter verständigen.

---

# 10 Anhang

Im Anhang finden Sie:

- die Privilegien bei POSIX
- den Kommandoumfang der Basis-Shell
- die Dateiverzeichnisse, die bei einer Erstinstallation angelegt werden
- die Gerätedateien, die bei einer Erstinstallation angelegt werden
- die Verwaltungsdateien, die bei einer Erstinstallation angelegt werden
- die SDF-Syntaxbeschreibung
- die Kommando-Returncodes

## 10.1 Privilegien bei POSIX

Die folgende Tabelle zeigt, wer welche Aufgaben in POSIX ausführen darf:

Privileg	Berechtigung für
(Privileg <b>SUBSYSTEM-MANAGEMENT</b> )	POSIX-Status anzeigen: /SHOW-POSIX-STATUS
	POSIX starten: /START-SUBSYSTEM SUBSYSTEM-NAME=POSIX
	POSIX beenden: /STOP-SUBSYSTEM SUBSYSTEM-NAME=POSIX
Sicherheitsbeauftragter  (Privileg <b>SECURITY-ADMINISTRATION</b> )	BS2000-Kennungen (außer SYSROOT) das Privileg POSIX-ADMINISTRATION verleihen oder entziehen: /SET-PRIVILEGE /RESET-PRIVILEGE
(Privileg <b>USER-ADMINISTRATION</b> , aber <b>keine</b> Root-Berechtigung)	Werte der Tuningparameter in der POSIX-Informationsdatei SYSSSI.POSIX-BC.010 ändern
	Neue POSIX-Benutzer verwalten: /ADD-USER und weitere Maßnahmen
	BS2000-Benutzerkennungen individuelle POSIX-Benutzer- attribute zuordnen: /MODIFY-POSIX-USER-ATTRIBUTES
	POSIX-Benutzerattribute verwalten: /MODIFY-POSIX-USER-ATTRIBUTES /SHOW-POSIX-USER-ATTRIBUTES
	Standardwerte für POSIX-Benutzerattribute verwalten: /MODIFY-POSIX-USER-DEFAULTS /SHOW-POSIX-USER-DEFAULTS
	Zugangsberechtigung für den Benutzer eines fernen Rechners verwalten: /SET-LOGON-PROTECTION /MODIFY-LOGON-PROTECTION /SHOW-LOGON-PROTECTION
	Abrechnungsnummer für den Systemzugang über einen fernen Rechner verwalten: /ADD-USER /MODIFY-USER-ATTRIBUTES /SHOW-USER-ATTRIBUTES
	POSIX-Gruppen/Benutzer verwalten: Benutzerattribut GROUP-NUMBER

Privileg	Berechtigung für
BS2000-Systemverwalter  (Privileg <b>USER-ADMINISTRATION</b> und <b>zusätzlich</b> Root-Berechtigung)	Alle Aufgaben, die ein BS2000-Systemverwalter ohne Root-Berechtigung durchführen darf, und zusätzlich:  POSIX installieren: POSIX-Installationsprogramm  POSIX-Dateisysteme einrichten, ändern und löschen: POSIX-Installationsprogramm
POSIX-Verwalter  (Privileg <b>POSIX-ADMINISTRATION</b> und zusätzlich Root-Berechtigung)	Neue POSIX-Benutzer verwalten: /ADD-USER und weitere Maßnahmen  POSIX-Benutzer löschen: /MODIFY-POSIX-USER-ATTRIBUTES  POSIX-Dateisysteme einrichten, ändern , löschen: POSIX-Installationsprogramm  POSIX-Dateisysteme ein- und aushängen: <i>mount</i> und <i>mountall</i> ; <i>umount</i> und <i>umountall</i>  POSIX-Benutzerattribute verwalten: /MODIFY-POSIX-USER-ATTRIBUTES /SHOW-POSIX-USER-ATTRIBUTES  Standardwerte für POSIX-Benutzerattribute verwalten: /MODIFY-POSIX-USER-DEFAULTS /SHOW-POSIX-USER-DEFAULTS  BS2000- und POSIX-Gruppen verwalten: Benutzerattribut GROUP-NUMBER, Datei <i>/etc/group</i>
Root-Berechtigung  <b>(Benutzernummer 0, Gruppennummer 0)</b>	POSIX-Gruppen in POSIX verwalten: Datei <i>/etc/group</i>  POSIX-Benutzer löschen: <i>rmdir</i>

Privileg	Berechtigung für
BS2000-Gruppenverwalter  (Gruppe <b>*UNIVERSAL</b> )	POSIX-Benutzerattribute verwalten (mit Einschränkungen): /MODIFY-POSIX-USER-ATTRIBUTES /SHOW-POSIX-USER-ATTRIBUTES
	Standardwerte für POSIX-Benutzerattribute verwalten (mit Einschränkungen): /MODIFY-POSIX-USER-DEFAULTS /SHOW-POSIX-USER-DEFAULTS
	Zugangsberechtigung für den Benutzer eines fernen Rechners verwalten (mit Einschränkungen): /SET-LOGON-PROTECTION /MODIFY-LOGON-PROTECTION /SHOW-LOGON-PROTECTION
	Abrechnungsnummer für den Systemzugang über einen fernen Rechner verwalten (mit Einschränkungen): /ADD-USER /MODIFY-USER-ATTRIBUTES /SHOW-USER-ATTRIBUTES
	POSIX-Gruppen/Benutzer verwalten: Benutzerattribut GROUP-NUMBER
Nichtprivilegierter POSIX-Benutzer  (Privileg <b>STD-PROCESSING</b> )	Informationen über die Einträge im Benutzerkatalog für die eigenen Benutzerkennungen ausgeben: /SHOW-USER-ATTRIBUTES /SHOW-POSIX-USER-ATTRIBUTES

## 10.2 Kommando umfang der Basis-Shell

Die POSIX-Shell (Basis-Shell), die Bestandteil von POSIX-BC ist, beinhaltet folgende POSIX-Kommandos:

POSIX-Kommando	Funktion
adduser	Individuelle Benutzernummer zuteilen
alias	Alias-Namen definieren oder anzeigen
ar	Bibliotheken verwalten
basename	Dateinamen vom Pfad trennen
bg	Jobs im Hintergrund bearbeiten
break	Schleife im Shell-Script abbrechen
bs2cp	BS2000-Dateien kopieren
bs2file	Dateiattribute für BS2000-Dateien festlegen
bs2lp	Dateien ausdrucken
bs2pkey	P-Tasten belegen
case	Anweisung (Shell-Script)
cat	Dateien aneinanderfügen und ausgeben
cd	Aktuelles Dateiverzeichnis wechseln
chgrp	Gruppennummer einer Datei ändern
chmod	Zugriffsrechte ändern
chown	Eigentümer einer Datei ändern
comm	Gleiche Zeilen in zwei sortierten Dateien suchen
command	Einfaches Kommando ausführen
continue	Schleifen steuern
cp	Dateien kopieren
date	Datum und Uhrzeit ausgeben
debug	Testen von POSIX-Programmen
df	Anzahl der freien und belegten Plattenblöcke und I-Nodes ausgeben
du	Belegten Speicherplatz ausgeben
dumpfs	Interne Dateisystem-Information ausgeben
echo	Aufruf-Argumente ausgeben
edt	BS2000-Dateibearbeiter EDT aufrufen
eval	Aufruf-Argumente bearbeiten und als Kommando ausführen

POSIX-Kommando	Funktion
exec	Aktuelle Shell überlagern
exit	Shell-Prozedur beenden
export	Shell-Variablen exportieren
expr	Ausdrücke auswerten
false	Endestatus ungleich 0 zurückgeben
fc	Zugriff auf die History-Datei
fg	Jobs in den Vordergrund bringen
for	Schleifen steuern
fsck	Konsistenz eines Dateisystems prüfen und Korrektur im Dialog mit dem Benutzer
ftyp	Bearbeitungsart für Dateien festlegen
getopts	Argumente einer Prozedur nach Optionen durchsuchen
grep	Muster suchen
hash	Hash-Tabelle der Shell bearbeiten
hd	Dateinhalt hexadezimal ausgeben
iconv	Code konvertieren
id	Benutzer-Identifikation ausgeben
if	Schleifen steuern
ipcrm	Einrichtungen zur Interprozeß-Kommunikation entfernen
ipcs	Zustand von Interprozeß-Kommunikationseinrichtungen ausgeben
info	Online-Diagnose-Tool
jobs	Auftragsinformationen ausgeben
kill	Signale an Prozesse senden
let	Integer-Arithmetik
ln	Verweis auf eine Datei eintragen
ls	Informationen über Dateiverzeichnisse und Dateien ausgeben
mkdir	Dateiverzeichnis erzeugen
mknod	Gerätedatei anlegen
mount	Dateisysteme einhängen
mountall	Mehrere Dateisysteme einhängen
mv	Dateien versetzen oder umbenennen
newgrp	Gruppenzugehörigkeit ändern
od	Dateinhalt oktal ausgeben

POSIX-Kommando	Funktion
posdbl	POSIX-Lader verwalten
print	Ausgabemechanismus ähnlich wie echo
ps	Prozeßdaten abfragen
pwd	Pfadnamen des aktuellen Dateiverzeichnisses ausgeben
rcp	Dateien zwischen POSIX-BS2000 und SINIX/UNIX kopieren
read	Argumente von der Standard-Eingabe lesen und Shell-Variablen zuweisen
readonly	Shell-Variablen schützen
return	Funktionen über Script abbrechen
rm	Dateien löschen
rmdir	Dateiverzeichnisse löschen
rsh	Kommandos auf einem SINIX-/UNIX-Rechner ausführen
set	Shell-Optionen oder Stellungsparameter setzen
sh	Komandointerpreter und Programmiersprache POSIX-Shell
shift	Werte der Stellungsparameter nach links verschieben
sleep	Prozesse zeitweise stilllegen
sort	Dateien sortieren und/oder mischen
stty	Eigenschaften einer Datensichtstations ausgeben oder ändern
sync	Systempuffer zurückschreiben
test	Bedingungen prüfen
time	Laufzeit eines Kommandos messen
times	Gesamlaufzeit der bisher gestarteten Prozesse ausgeben
touch	Änderungs- und Zugriffszeiten aktualisieren
tr	Zeichen ersetzen oder löschen
trap	Signalbehandlung ändern
true	Endestatus 0 zurückgeben
type	Typ eines Kommandos abfragen
typeset	Attribute für Shell-Variable setzen
ulimit	Dateigröße für das Schreiben begrenzen oder aktuellen Grenzwert abfragen
umask	Standard-Vergabe der Zugriffsrechte ausgeben oder ändern
umount	Dateisysteme aushängen
umountall	Mehrere Dateisysteme aushängen
unalias	Variablen aus der alias-Tabelle löschen

POSIX-Kommando	Funktion
uname	Basisdaten über das aktuelle Betriebssystem ausgeben
uniq	Mehrfache Zeilen suchen
unset	Shell-Variablen oder Shell-Funktionen aus der Umgebung löschen
until	Schleifen steuern
wait	Auf die Beendigung von Hintergrundprozessen warten
whence	Kommando-Lokalisation
while	Schleifen steuern

## 10.3 Dateiverzeichnisse, die bei einer Erstinstallation angelegt werden

Die folgenden Dateiverzeichnisse werden bei einer Erstinstallation angelegt:

**/dev** mit den Unterverzeichnissen:

/dsk, /fd, /pts, /rds, /sad, /sf und /term

**/etc** mit den Unterverzeichnissen:

/default, /dfs, /fs, /inet, /init.d, /net, /net/ticlts, /net/ticots, /net/ticotsord, /sm, /sm.d, ./products und ./products/.legit

**/home**

**/lost+found**

**/proc**

**/sbin**

**/tmp**

**/usr** mit den Unterverzeichnissen:

/bin, /lib, /lib/iconv, /lib/lex, /lib/nfs, /sbin und /ucb

**/var** mit den Unterverzeichnissen:

/adm, /lp, /mail, /preserve, /spool, /spool/lp/temp, /sadm/pkg und /tmp

Diese Dateiverzeichnisse, die das POSIX-Installationsprogramm einrichtet, darf der POSIX-Verwalter weder editieren noch verändern!

## 10.4 Gerätedateien, die bei einer Erstinstallation angelegt werden

Die folgenden Gerätedateien werden bei einer Erstinstallation angelegt:

**/dev/console**

**/dev/kmem**

**/dev/log**

**/dev/loop**

**/dev/null**

**/dev/dsk/0s0**

**/dev/rdsk/0s0**

**/dev/root**

**/dev/rroot**

**/dev/ptmx**

**/dev/sf/mmm**      000 <= mmm < 256

**/dev/ticlts**

**/dev/ticots**

**/dev/ticotsord**

**/dev/tty**

**/dev/zero**

**/dev/sad/admin**

**/dev/sad/user**

**/dev/ptmx**

**/dev/pts/mmm**      000 <= mmm < 256

**/dev/term/mmm**      000 <= mmm < 256

Diese Gerätedateien, die das POSIX-Installationsprogramm einrichtet, darf der POSIX-Verwalter weder editieren noch verändern!

## 10.5 Verwaltungsdateien, die bei einer Erstinstallation angelegt werden

Die folgenden Verwaltungsdateien werden bei einer Erstinstallation angelegt:

**/etc/dfs/dfstab**  
**/etc/dfs/fstypes**  
**/etc/dfs/sharetab**  
**/etc/group**  
**/etc/inetd.conf**  
**/etc/mnttab**  
**/etc/net/ticlts/hosts**  
**/etc/net/ticlts/services**  
**/etc/net/ticots/hosts**  
**/etc/net/ticots/services**  
**/etc/net/ticotsord/hosts**  
**/etc/net/ticotsord/services**  
**/etc/netconfig**  
**/etc/partitions**  
**/etc/print**  
**/etc/protocols**  
**/etc/profile**  
**/etc/services**  
**/etc/termcap**  
**/etc/TIMEZONE**  
**/etc/vfstab**

Die Tabellen, die in */etc* eingerichtet werden, stimmen in Bezug auf Inhalt und Bedeutung mit den entsprechenden Tabellen von SINIX V5.41 überein.

## 10.6 Tuning-Maßnahmen

Die folgende Tabelle enthält Maßnahmen, die zur Steigerung der POSIX-Performance empfohlen werden.

Maßnahme	Bemerkungen
Verwendung des POSIX-Laders	<p>Empfohlen wird ein Memory-Pool mit 30 MB (Parameter DBLPOOL in der Informationsdatei)</p> <p>Hinweis: Die PAGING AREA muß diesen zusätzlichen Adreßraum aufnehmen können.</p>
Einsatz aktueller Subsysteme	<p>Für die Shell-Kommandos und die Ausführung eigener Programme sollten folgende Subsysteme gestartet sein:</p> <p>Für CRTE V2.0: CRTEC</p> <p style="padding-left: 40px;">CRTECOM CRTEBASY (für /390-Anlagen) CRTEBASR (für RISC-Anlagen)</p> <p>Für CRTE V2.1: CRTEC</p> <p style="padding-left: 40px;">CRTEPART CRTEBASY (für /390-Anlagen) CRTEBASR (für RISC-Anlagen)</p> <p>Für die Nutzung von EDT: EDT und EDTCON</p> <p>Für die Programmproduktion: BINDER, PMLOG</p>
Nutzung von DAB	<p>Folgende Dateien sollten im Cache gehalten werden:</p> <p>SYS.AIDIT0 (ca. 30 PAM-Seiten groß): vollständig cachen</p> <p>Für die Programmproduktion:</p> <p>SINLIB.&lt;compiler&gt;.&lt;version&gt; (je nach Nutzung bis zu mehreren MB)</p>

## 10.7 SDF-Syntaxbeschreibung

Diese Syntaxbeschreibung basiert auf der SDF-Version 4.1A. Die Syntax der SDF-Kommando-/Anweisungssprache wird im folgenden in drei Tabellen erklärt.

*Zu Tabelle 1: Metasyntax*

In den Kommando-/Anweisungsformaten werden bestimmte Zeichen und Darstellungsformen verwendet, deren Bedeutung in Tabelle 1 erläutert wird.

*Zu Tabelle 2: Datentypen*

Variable Operandenwerte werden in SDF durch Datentypen dargestellt. Jeder Datentyp repräsentiert einen bestimmten Wertevorrat. Die Anzahl der Datentypen ist beschränkt auf die in Tabelle 2 beschriebenen Datentypen.

Die Beschreibung der Datentypen gilt für alle Kommandos und Anweisungen. Deshalb werden bei den entsprechenden Operandenbeschreibungen nur noch Abweichungen von Tabelle 2 erläutert.

*Zu Tabelle 3: Zusätze zu Datentypen*

Zusätze zu Datentypen kennzeichnen weitere Eingabevorschriften für Datentypen. Die Zusätze enthalten eine Längen- bzw. Intervallangabe, schränken den Wertevorrat ein (Zusatz beginnt mit *without*), erweitern ihn (Zusatz beginnt mit *with*) oder erklären eine bestimmte Angabe zur Pflichtangabe (Zusatz beginnt mit *mandatory*). Im Handbuch werden folgende Zusätze in gekürzter Form dargestellt:

cat-id	cat
completion	compl
correction-state	corr
generation	gen
lower-case	low
manual-release	man
odd-possible	odd
path-completion	path-compl
separators	sep
temporary-file	temp-file
underscore	under
user-id	user
version	vers
wildcard-constr	wild-constr
wildcards	wild

Für den Datentyp `integer` enthält Tabelle 3 außerdem kursiv gesetzte Einheiten, die nicht Bestandteil der Syntax sind. Sie dienen lediglich als Lesehilfe.

Für Sonderdatentypen, die durch die Implementierung geprüft werden, enthält Tabelle 3 kursiv gesetzte Zusätze (siehe Zusatz *special*), die nicht Bestandteil der Syntax sind.

Die Beschreibung der Zusätze zu den Datentypen gilt für alle Kommandos und Anweisungen. Deshalb werden bei den entsprechenden Operandenbeschreibungen nur noch Abweichungen von Tabelle 3 erläutert.

## Metasyntax

Kennzeichnung	Bedeutung	Beispiele
<b>GROSSBUCHSTABEN</b>	Großbuchstaben kennzeichnen Schlüsselwörter (Kommando-, Anweisungs-, Operandennamen, Schlüsselwortwerte) und konstante Operandenwerte. Schlüsselwortwerte beginnen mit *	<b>HELP-SDF</b>  <b>SCREEN-STEPS = *NO</b>
<b>GROSSBUCHSTABEN</b> in Halbfett	Großbuchstaben in Halbfett kennzeichnen garantierte bzw. vorgeschlagene Abkürzungen der Schlüsselwörter.	<b>GUIDANCE-MODE = *YES</b>
=	Das Gleichheitszeichen verbindet einen Operandennamen mit den dazugehörenden Operandenwerten.	<b>GUIDANCE-MODE = *NO</b>
< >	Spitze Klammern kennzeichnen Variablen, deren Wertevorrat durch Datentypen und ihre Zusätze beschrieben wird (siehe Tabellen 2 und 3).	<b>SYNTAX-FILE = &lt;filename 1..54&gt;</b>
<u>Unterstreichung</u>	Der Unterstrich kennzeichnet den Default-Wert eines Operanden.	<b>GUIDANCE-MODE = *NO</b>
/	Der Schrägstrich trennt alternative Operandenwerte.	<b>NEXT-FIELD = *NO / *YES</b>
(...)	Runde Klammern kennzeichnen Operandenwerte, die eine Struktur einleiten.	<b>,UNGUIDED-DIALOG = *YES (...) / *NO</b>

Tabelle 1: Metasyntax (Teil 1 von 2)

Tabelle 1: Metasyntax (Teil 2 von 2)

## Datentypen

Datentyp	Zeichenvorrat	Besonderheiten
alphanum-name	A...Z 0...9 \$, #, @	
cat-id	A...Z 0...9	maximal 4 Zeichen; darf nicht mit der Zeichenfolge PUB beginnen
command-rest	beliebig	
composed-name	A...Z 0...9 \$, #, @ Bindestrich Punkt Katalogkennung	alphanumerische Zeichenfolge, die in mehrere durch Punkt oder Bindestrich getrennte Teilzeichenfolgen gegliedert sein kann. Ist auch die Angabe eines Dateinamens möglich, so kann die Zeichenfolge mit einer Katalogkennung im Format :cat: beginnen (siehe Datentyp filename).
c-string	EBCDIC-Zeichen	ist in Hochkommata einzuschließen; der Buchstabe C kann vorangestellt werden; Hochkommata innerhalb des c-string müssen verdoppelt werden
date	0...9 Strukturkennzeichen: Bindestrich	Eingabeformat: jjjj-mm-tt  jjjj: Jahr; wahlweise 2- oder 4stellig mm: Monat tt: Tag
device	A...Z 0...9 Bindestrich	Zeichenfolge, die maximal 8 Zeichen lang ist und einem im System verfügbaren Gerät entspricht. In der Dialogführung zeigt SDF die zulässigen Operandenwerte an. Hinweise zu möglichen Geräten sind der jeweiligen Operandenbeschreibung zu entnehmen.
fixed	+, - 0...9 Punkt	Eingabeformat: [zeichen][ziffern].[ziffern]  [zeichen]: + oder - [ziffern]: 0...9  muß mindestens eine Ziffer, darf aber außer dem Vorzeichen maximal 10 Zeichen (0...9, Punkt) enthalten

Tabelle 2: Datentypen (Teil 1 von 6)

Datentyp	Zeichenvorrat	Besonderheiten
filename	A...Z 0...9 \$, #, @ Bindestrich Punkt	<p>Eingabeformat:</p> $[:cat:][\$user.] \left\{ \begin{array}{l} \text{datei} \\ \text{datei(nr)} \\ \text{gruppe} \\ \text{gruppe} \left\{ \begin{array}{l} (*\text{abs}) \\ (+\text{rel}) \\ (-\text{rel}) \end{array} \right. \end{array} \right\}$ <p>:cat: wahlfreie Angabe der Katalogkennung; Zeichenvorrat auf A...Z und 0...9 eingeschränkt; max. 4 Zeichen; ist in Doppelpunkte einzuschließen; voreingestellt ist die Katalogkennung, die der Benutzerkennung laut Eintrag im Benutzerkatalog zugeordnet ist.</p> <p>\$user. wahlfreie Angabe der Benutzerkennung; Zeichenvorrat ist A...Z, 0...9, \$, #, @; max. 8 Zeichen; darf nicht mit einer Ziffer beginnen; \$ und Punkt müssen angegeben werden; voreingestellt ist die eigene Benutzerkennung.</p> <p>\$. (Sonderfall) System-Standardkennung</p> <p>datei Datei- oder Jobvariablenname; kann durch Punkt in mehrere Teilnamen gegliedert sein: name<sub>1</sub>[.name<sub>2</sub>[...]] name<sub>i</sub> enthält keinen Punkt und darf nicht mit Bindestrich beginnen oder enden; datei ist max. 41 Zeichen lang, darf nicht mit \$ beginnen und muß mindestens ein Zeichen aus A...Z enthalten.</p>

Tabelle 2: Datentypen (Teil 2 von 6)

Datentyp	Zeichenvorrat	Besonderheiten
filename (Forts.)		<p>#datei (Sonderfall)  @datei (Sonderfall)  # oder @ als erstes Zeichen kennzeichnet je nach Systemparameter temporäre Dateien und Jobvariablen.</p> <p>datei(nr)  Banddateiname  nr: Versionsnummer;  Zeichenvorrat ist A...Z, 0...9, \$, #, @.  Klammern müssen angegeben werden.</p> <p>gruppe  Name einer Dateigenerationsgruppe  (Zeichenvorrat siehe unter "datei")</p> <p>gruppe <math>\left\{ \begin{array}{l} (*\text{abs}) \\ (+\text{rel}) \\ (-\text{rel}) \end{array} \right\}</math></p> <p>(*abs)  absolute Generationsnummer (1..9999);  * und Klammern müssen angegeben werden.</p> <p>(+rel)  (-rel)  relative Generationsnummer (0..99);  Vorzeichen und Klammern müssen angegeben werden.</p>
integer	0...9, +, -	+ bzw. - kann nur erstes Zeichen sein (Vorzeichen).
name	A...Z 0...9 \$, #, @	darf nicht mit einer Ziffer beginnen.

Tabelle 2: Datentypen (Teil 3 von 6)

Datentyp	Zeichenvorrat	Besonderheiten
partial-filename	A...Z 0...9 \$, #, @ Bindestrich Punkt	Eingabeformat: [:cat:][\$user.][teilname.]  :cat: siehe filename \$user. siehe filename  teilname wahlfreie Angabe des gemeinsamen ersten Namensteils von Dateien und Dateigenera- tionsgruppen in der Form: name <sub>1</sub> .[name <sub>2</sub> .[...]] name <sub>i</sub> siehe filename. Das letzte Zeichen von teilname muß ein Punkt sein. Es muß mindestens einer der Teile :cat:, \$user. oder teilname angegeben werden.
posix-filename	A...Z 0...9 Sonderzeichen	Zeichenfolge, die maximal 255 Zeichen lang ist. Besteht entweder aus einem oder zwei Punk- ten, oder aus alphanumerischen Zeichen und Sonderzeichen; Sonderzeichen sind mit dem Zeichen \ zu entwerten. Nicht erlaubt ist das Zei- chen /. Muß in Hochkommata eingeschlossen werden, wenn alternative Datentypen zulässig sind, Se- paratoren verwendet werden oder das erste Zeichen ?, ! bzw. ^ ist. Zwischen Groß- und Kleinschreibung wird un- terschieden.
posix-pathname	A...Z 0...9 Sonderzeichen Strukturkennzeichen: Schrägstrich	Eingabeformat: [/]part <sub>1</sub> [/.../part <sub>i</sub> ] wobei part <sub>i</sub> ein posix-filename ist; maximal 1023 Zeichen; muß in Hochkommata eingeschlossen werden, wenn alternative Datentypen zulässig sind, Separatoren verwendet werden oder das erste Zeichen ?, ! bzw. ^ ist.

Tabelle 2: Datentypen (Teil 4 von 6)

Datentyp	Zeichenvorrat	Besonderheiten
product-version	A...Z 0...9 Punkt Hochkomma	<p>Eingabeformat: <math>[[C]' ][V][m]m.nasol[' ]</math></p> <p style="text-align: right;">↓ Korrekturstand Freigabestand</p> <p>wobei m, n, s und o jeweils eine Ziffer und a ein Buchstabe ist.</p> <p>Ob Freigabe- und/oder Korrekturstand angegeben werden dürfen oder ob sie angegeben werden müssen, bestimmen Zusätze zu dem Datentyp (siehe Tabelle 3, Zusätze without-corr, without-man, mandatory-man und mandatory-corr).</p> <p>product-version kann in Hochkommata eingeschlossen werden, wobei der Buchstabe C vorangestellt werden kann. Die Versionsangabe kann mit dem Buchstaben V beginnen.</p>
structured-name	A...Z 0...9 \$, #, @ Bindestrich	alphanumerische Zeichenfolge, die in mehrere durch Bindestrich getrennte Teilzeichenfolgen gegliedert sein kann; erstes Zeichen: A...Z oder \$, #, @
text	beliebig	Das Eingabeformat ist den jeweiligen Operandenbeschreibungen zu entnehmen.
time	0...9 Strukturkennzeichen: Doppelpunkt	<p>Angabe einer Tageszeit</p> <p>Eingabeformat: <math>\left\{ \begin{array}{l} hh:mm:ss \\ hh:mm \\ hh \end{array} \right\}</math></p> <p>hh: Stunden mm: Minuten ss: Sekunden</p> <p>führende Nullen können weggelassen werden</p>
vsn	a) A...Z 0...9  b) A...Z 0...9 \$, #, @	a) Eingabeformat: pvsid.folgenummer max. 6 Zeichen; pvsid: 2-4 Zeichen; Eingabe von PUB nicht erlaubt folgenummer: 1-3 Zeichen  b) max. 6 Zeichen; PUB darf vorangestellt werden, dann dürfen jedoch nicht \$, #, @ folgen.

Tabelle 2: Datentypen (Teil 5 von 6)

<b>Datentyp</b>	<b>Zeichenvorrat</b>	<b>Besonderheiten</b>
x-string	Sedezimal: 00...FF	ist in Hochkommata einzuschließen; der Buchstabe X muß vorangestellt werden; die Anzahl der Zeichen darf ungerade sein.
x-text	Sedezimal: 00...FF	ist nicht in Hochkommata einzuschließen; der Buchstabe X darf nicht vorangestellt werden; die Anzahl der Zeichen darf ungerade sein.

Tabelle 2: Datentypen (Teil 6 von 6)

## Zusätze zu Datentypen

Zusatz	Bedeutung										
<i>x..y unit</i>	<p>beim Datentyp integer: Intervallangabe</p> <p>x Mindestwert, der für integer erlaubt ist. x ist eine ganze Zahl, die mit einem Vorzeichen versehen werden darf.</p> <p>y Maximalwert, der für integer erlaubt ist. y ist eine ganze Zahl, die mit einem Vorzeichen versehen werden darf.</p> <p><i>unit</i> nur bei Datentyp integer: zusätzliche Einheiten. Folgende Angaben werden verwendet:</p> <table> <tr><td><i>days</i></td><td><i>byte</i></td></tr> <tr><td><i>hours</i></td><td><i>2Kbyte</i></td></tr> <tr><td><i>minutes</i></td><td><i>4Kbyte</i></td></tr> <tr><td><i>seconds</i></td><td><i>Mbyte</i></td></tr> </table>	<i>days</i>	<i>byte</i>	<i>hours</i>	<i>2Kbyte</i>	<i>minutes</i>	<i>4Kbyte</i>	<i>seconds</i>	<i>Mbyte</i>		
<i>days</i>	<i>byte</i>										
<i>hours</i>	<i>2Kbyte</i>										
<i>minutes</i>	<i>4Kbyte</i>										
<i>seconds</i>	<i>Mbyte</i>										
<i>x..y special</i>	<p>bei den übrigen Datentypen: Längenangabe</p> <p>Bei den Datentypen catid, date, device, product-version, time und vsn wird die Längenangabe nicht angezeigt.</p> <p>x Mindestlänge für den Operandenwert; x ist eine ganze Zahl.</p> <p>y Maximallänge für den Operandenwert; y ist eine ganze Zahl.</p> <p>x=y Der Operandenwert muß genau die Länge x haben.</p> <p><i>special</i> Zusatzangabe zur Beschreibung eines Sonderdatentyps, der durch die Implementierung geprüft wird. Vor special können weitere Zusätze stehen. Folgende Angaben werden verwendet:</p> <table> <tr><td><i>arithm-expr</i></td><td>arithmetischer Ausdruck (SDF-P)</td></tr> <tr><td><i>bool-expr</i></td><td>logischer Ausdruck (SDF-P)</td></tr> <tr><td><i>string-expr</i></td><td>String-Ausdruck (SDF-P)</td></tr> <tr><td><i>expr</i></td><td>beliebiger Ausdruck (SDF-P)</td></tr> <tr><td><i>cond-expr</i></td><td>bedingter Ausdruck (JV)</td></tr> </table>	<i>arithm-expr</i>	arithmetischer Ausdruck (SDF-P)	<i>bool-expr</i>	logischer Ausdruck (SDF-P)	<i>string-expr</i>	String-Ausdruck (SDF-P)	<i>expr</i>	beliebiger Ausdruck (SDF-P)	<i>cond-expr</i>	bedingter Ausdruck (JV)
<i>arithm-expr</i>	arithmetischer Ausdruck (SDF-P)										
<i>bool-expr</i>	logischer Ausdruck (SDF-P)										
<i>string-expr</i>	String-Ausdruck (SDF-P)										
<i>expr</i>	beliebiger Ausdruck (SDF-P)										
<i>cond-expr</i>	bedingter Ausdruck (JV)										
<i>with</i>	<p>Erweitert die Angabemöglichkeiten für einen Datentyp.</p> <p><i>-compl</i> Bei Angaben zu dem Datentyp date ergänzt SDF zweistellige Jahresangaben der Form jj-mm-tt zu:</p> <table> <tr><td>20jj-mm-tt</td><td>falls jj &lt; 60</td></tr> <tr><td>19jj-mm-tt</td><td>falls jj ≥ 60</td></tr> </table> <p><i>-low</i> Groß- und Kleinschreibung wird unterschieden.</p> <p><i>-path-compl</i> Bei Angaben zu dem Datentyp filename ergänzt SDF die Katalog- und/oder die Benutzerkennung, falls diese nicht angegeben werden.</p> <p><i>-under</i> Erlaubt Unterstriche '_' beim Datentyp name.</p>	20jj-mm-tt	falls jj < 60	19jj-mm-tt	falls jj ≥ 60						
20jj-mm-tt	falls jj < 60										
19jj-mm-tt	falls jj ≥ 60										

Tabelle 3: Zusätze zu Datentypen (Teil 1 von 7)

Zusatz	Bedeutung										
with (Forts.) -wild(n)	<p>Teile eines Namens dürfen durch die folgenden Platzhalter ersetzt werden. n bezeichnet die maximale Eingabelänge bei Verwendung von Platzhaltern. Mit Einführung der Datentypen posix-filename und posix-pathname akzeptiert SDF neben den bisher im BS2000 üblichen Platzhaltern auch Platzhalter aus der UNIX-Welt (nachfolgend POSIX-Platzhalter genannt). Da derzeit nicht alle Kommandos POSIX-Platzhalter unterstützen, kann ihre Verwendung bei Datentypen ungleich posix-filename und posix-pathname zu Semantikfehlern führen.</p> <p>Innerhalb einer Musterzeichenfolge sollten entweder nur BS2000- oder nur POSIX-Platzhalter verwendet werden. Bei den Datentypen posix-filename und posix-pathname sind nur POSIX-Platzhalter erlaubt. Ist eine Musterzeichenfolge mehrdeutig auf einen String abbildbar, gilt der erste Treffer.</p>										
BS2000- Platzhalter	<table border="1"> <thead> <tr> <th></th> <th>Bedeutung</th> </tr> </thead> <tbody> <tr> <td>*</td><td>Ersetzt eine beliebige, auch leere Zeichenfolge. Ein * an erster Stelle muß verdoppelt werden, sofern dem * weitere Zeichen folgen und die eingegebene Zeichenfolge nicht mindestens einen weiteren Platzhalter enthält.</td></tr> <tr> <td>Punkt am Ende</td><td>Teilqualifizierte Angabe eines Namens. Entspricht implizit der Zeichenfolge "./*", d.h. nach dem Punkt folgt mindestens ein beliebiges Zeichen.</td></tr> <tr> <td>/</td><td>Ersetzt genau ein beliebiges Zeichen.</td></tr> <tr> <td>&lt;s<sub>x</sub>:s<sub>y</sub>&gt;</td><td>Ersetzt eine Zeichenfolge, für die gilt: <ul style="list-style-type: none"> <li>– sie ist mindestens so lang wie die kürzeste Zeichenfolge (s<sub>x</sub> oder s<sub>y</sub>)</li> <li>– sie ist höchstens so lang wie die längste Zeichenfolge (s<sub>x</sub> oder s<sub>y</sub>)</li> <li>– sie liegt in der alphabetischen Sortierung zwischen s<sub>x</sub> und s<sub>y</sub>; Zahlen werden hinter Buchstaben sortiert (A...Z 0...9)</li> <li>– s<sub>x</sub> darf auch die leere Zeichenfolge sein, die in der alphabetischen Sortierung an erster Stelle steht</li> <li>– s<sub>y</sub> darf auch die leere Zeichenfolge sein, die an dieser Stelle für die Zeichenfolge mit der höchst möglichen Codierung steht (enthält nur die Zeichen X' FF' )</li> </ul> </td></tr> </tbody> </table>		Bedeutung	*	Ersetzt eine beliebige, auch leere Zeichenfolge. Ein * an erster Stelle muß verdoppelt werden, sofern dem * weitere Zeichen folgen und die eingegebene Zeichenfolge nicht mindestens einen weiteren Platzhalter enthält.	Punkt am Ende	Teilqualifizierte Angabe eines Namens. Entspricht implizit der Zeichenfolge "./*", d.h. nach dem Punkt folgt mindestens ein beliebiges Zeichen.	/	Ersetzt genau ein beliebiges Zeichen.	<s <sub>x</sub> :s <sub>y</sub> >	Ersetzt eine Zeichenfolge, für die gilt: <ul style="list-style-type: none"> <li>– sie ist mindestens so lang wie die kürzeste Zeichenfolge (s<sub>x</sub> oder s<sub>y</sub>)</li> <li>– sie ist höchstens so lang wie die längste Zeichenfolge (s<sub>x</sub> oder s<sub>y</sub>)</li> <li>– sie liegt in der alphabetischen Sortierung zwischen s<sub>x</sub> und s<sub>y</sub>; Zahlen werden hinter Buchstaben sortiert (A...Z 0...9)</li> <li>– s<sub>x</sub> darf auch die leere Zeichenfolge sein, die in der alphabetischen Sortierung an erster Stelle steht</li> <li>– s<sub>y</sub> darf auch die leere Zeichenfolge sein, die an dieser Stelle für die Zeichenfolge mit der höchst möglichen Codierung steht (enthält nur die Zeichen X' FF' )</li> </ul>
	Bedeutung										
*	Ersetzt eine beliebige, auch leere Zeichenfolge. Ein * an erster Stelle muß verdoppelt werden, sofern dem * weitere Zeichen folgen und die eingegebene Zeichenfolge nicht mindestens einen weiteren Platzhalter enthält.										
Punkt am Ende	Teilqualifizierte Angabe eines Namens. Entspricht implizit der Zeichenfolge "./*", d.h. nach dem Punkt folgt mindestens ein beliebiges Zeichen.										
/	Ersetzt genau ein beliebiges Zeichen.										
<s <sub>x</sub> :s <sub>y</sub> >	Ersetzt eine Zeichenfolge, für die gilt: <ul style="list-style-type: none"> <li>– sie ist mindestens so lang wie die kürzeste Zeichenfolge (s<sub>x</sub> oder s<sub>y</sub>)</li> <li>– sie ist höchstens so lang wie die längste Zeichenfolge (s<sub>x</sub> oder s<sub>y</sub>)</li> <li>– sie liegt in der alphabetischen Sortierung zwischen s<sub>x</sub> und s<sub>y</sub>; Zahlen werden hinter Buchstaben sortiert (A...Z 0...9)</li> <li>– s<sub>x</sub> darf auch die leere Zeichenfolge sein, die in der alphabetischen Sortierung an erster Stelle steht</li> <li>– s<sub>y</sub> darf auch die leere Zeichenfolge sein, die an dieser Stelle für die Zeichenfolge mit der höchst möglichen Codierung steht (enthält nur die Zeichen X' FF' )</li> </ul>										

Tabelle 3: Zusätze zu Datentypen (Teil 2 von 7)

Zusatz	Bedeutung	
with-wild(n) (Forts.)	< $s_1, \dots$ >	Ersetzt alle Zeichenfolgen, auf die eine der mit $s$ angegebenen Zeichenkombinationen zutrifft. $s$ kann auch die leere Zeichenfolge sein. Jede Zeichenfolge $s$ kann auch eine Bereichsangabe " $s_x:s_y$ " sein (siehe oben).
	- $s$	Ersetzt alle Zeichenfolgen, die der angegebenen Zeichenfolge $s$ nicht entsprechen. Das Minuszeichen darf nur am Beginn der Zeichenfolge stehen. Innerhalb der Datentypen filename bzw. partial-filename kann die negierte Zeichenfolge - $s$ genau einmal verwendet werden, d.h., - $s$ kann einen der drei Namensteile cat, user oder datei ersetzen.
Platzhalter sind in Generations- und Versionsangaben von Dateinamen nicht erlaubt. In Benutzerkennungen ist die Angabe von Platzhaltern der Systemverwaltung vorbehalten. Platzhalter können nicht die Begrenzer der Namensteile cat (Doppelpunkte) und user (\$) und Punkt ersetzen.		
POSIX- Platzhalter		Bedeutung
*		Ersetzt eine beliebige, auch leere Zeichenfolge. Ein * an erster Stelle muß verdoppelt werden, sofern dem * weitere Zeichen folgen und die eingegebene Zeichenfolge nicht mindestens einen weiteren Platzhalter enthält.
?		Ersetzt genau ein beliebiges Zeichen. Ist als erstes Zeichen außerhalb von Hochkommata nicht zulässig.
[ $c_x-c_y$ ]		Ersetzt genau ein Zeichen aus dem Bereich $c_x$ und $c_y$ einschließlich der Bereichsgrenzen. $c_x$ und $c_y$ müssen einfache Zeichen sein.
[ $s$ ]		Ersetzt genau ein Zeichen aus der Zeichenfolge $s$ . Die Ausdrücke [ $c_x-c_y$ ] und [ $s$ ] können kombiniert werden zu [ $s_1c_x-c_ys_2$ ]
[! $c_x-c_y$ ]		Ersetzt genau ein Zeichen, das nicht in dem Bereich $c_x$ und $c_y$ einschließlich der Bereichsgrenzen enthalten ist. $c_x$ und $c_y$ müssen einfache Zeichen sein. Die Ausdrücke [! $c_x-c_y$ ] und [ $s$ ] können kombiniert werden zu [! $s_1c_x-c_ys_2$ ]
[! $s$ ]		Ersetzt genau ein Zeichen, das nicht in der Zeichenfolge $s$ enthalten ist. Die Ausdrücke [! $s$ ] und [! $c_x-c_y$ ] können kombiniert werden zu [! $s_1c_x-c_ys_2$ ]

Tabelle 3: Zusätze zu Datentypen (Teil 3 von 7)

Zusatz	Bedeutung										
with (Forts.) -wild- constr(n)	<p>Angabe einer Konstruktionszeichenfolge, die angibt, wie aus einer zuvor angegebenen Auswahlzeichenfolge mit Musterzeichen (siehe with-wild) neue Namen zu bilden sind. n bezeichnet die maximale Eingabelänge bei Verwendung von Platzhaltern.</p> <p>Die Konstruktionszeichenfolge kann aus konstanten Zeichenfolgen und Musterzeichen bestehen. Ein Musterzeichen wird durch diejenige Zeichenfolge ersetzt, die durch das entsprechende Musterzeichen in der Auswahlzeichenfolge ausgewählt wird.</p> <p>Folgende Platzhalter können zur Konstruktionsangabe verwendet werden:</p> <table border="1"> <thead> <tr> <th>Platzhalter</th> <th>Bedeutung</th> </tr> </thead> <tbody> <tr> <td>*</td> <td>Entspricht der Zeichenfolge, die durch den Platzhalter * in der Auswahlzeichenfolge ausgewählt wird.</td> </tr> <tr> <td>Punkt am Ende</td> <td>Entspricht der teilqualifizierten Angabe eines Namens in der Auswahlzeichenfolge. Entspricht der Zeichenfolge, die durch den Punkt am Ende der Auswahlzeichenfolge ausgewählt wird.</td> </tr> <tr> <td>/ oder ?</td> <td>Entspricht dem Zeichen, das durch den Platzhalter / oder ? in der Auswahlzeichenfolge ausgewählt wird.</td> </tr> <tr> <td>&lt;n&gt;</td> <td>Entspricht der Zeichenfolge, die durch den n-ten Platzhalter in der Auswahlzeichenfolge ausgewählt wird; n = &lt;integer&gt;</td> </tr> </tbody> </table> <p>Zuordnung der Platzhalter zu entsprechenden Platzhaltern in der Auswahlzeichenfolge:</p> <p>In der Auswahlzeichenfolge werden alle Platzhalter von links nach rechts aufsteigend numeriert (globaler Index).</p> <p>Gleiche Platzhalter in der Auswahlzeichenfolge werden zusätzlich von links nach rechts aufsteigend numeriert (platzhalter-spezifischer Index).</p> <p>In der Konstruktionsangabe können Platzhalter auf zwei, sich gegenseitig ausschließende Arten angegeben werden:</p> <ol style="list-style-type: none"> <li>1. Platzhalter werden über den globalen Index angegeben: &lt;n&gt;</li> <li>2. Angabe desselben Platzhalters, wobei die Ersetzung gemäß dem platzhalter-spezifischen Index entsprechend erfolgt: z.B. der zweite "/" entspricht der Zeichenfolge, die durch den zweiten "/" in der Auswahlzeichenfolge ausgewählt wird.</li> </ol>	Platzhalter	Bedeutung	*	Entspricht der Zeichenfolge, die durch den Platzhalter * in der Auswahlzeichenfolge ausgewählt wird.	Punkt am Ende	Entspricht der teilqualifizierten Angabe eines Namens in der Auswahlzeichenfolge. Entspricht der Zeichenfolge, die durch den Punkt am Ende der Auswahlzeichenfolge ausgewählt wird.	/ oder ?	Entspricht dem Zeichen, das durch den Platzhalter / oder ? in der Auswahlzeichenfolge ausgewählt wird.	<n>	Entspricht der Zeichenfolge, die durch den n-ten Platzhalter in der Auswahlzeichenfolge ausgewählt wird; n = <integer>
Platzhalter	Bedeutung										
*	Entspricht der Zeichenfolge, die durch den Platzhalter * in der Auswahlzeichenfolge ausgewählt wird.										
Punkt am Ende	Entspricht der teilqualifizierten Angabe eines Namens in der Auswahlzeichenfolge. Entspricht der Zeichenfolge, die durch den Punkt am Ende der Auswahlzeichenfolge ausgewählt wird.										
/ oder ?	Entspricht dem Zeichen, das durch den Platzhalter / oder ? in der Auswahlzeichenfolge ausgewählt wird.										
<n>	Entspricht der Zeichenfolge, die durch den n-ten Platzhalter in der Auswahlzeichenfolge ausgewählt wird; n = <integer>										

Tabelle 3: Zusätze zu Datentypen (Teil 4 von 7)

Zusatz	Bedeutung
with-wild-constr (Forts.)	<p>Bei Konstruktionsangaben sind folgende Regeln zu beachten:</p> <ul style="list-style-type: none"> <li>– Die Konstruktionsangabe kann nur Platzhalter der Auswahlzeichenfolge enthalten.</li> <li>– Soll die Zeichenkette, die der Platzhalter &lt;...&gt; bzw. [...] auswählt, in der Konstruktionsangabe verwendet werden, muß die Index-Schreibweise gewählt werden.</li> <li>– Die Index-Schreibweise muß gewählt werden, wenn die Zeichenkette, die ein Platzhalter der Auswahlzeichenfolge bezeichnet, in der Konstruktionsangabe mehrfach verwendet werden soll: Bei der Auswahlangabe "A/" muß z.B. statt "A///" die Konstruktionszeichenfolge "A&lt;n&gt;&lt;n&gt;" angegeben werden.</li> <li>– Der Platzhalter * kann auch die leere Zeichenkette sein. Insbesondere ist zu beachten, daß bei mehreren Sternen in Folge (auch mit weiteren Platzhaltern) nur der letzte Stern eine nicht leere Zeichenfolge sein kann: z.B. bei "****" oder "*//*".</li> <li>– Aus der Konstruktionsangabe sollten gültige Namen entstehen. Darauf ist sowohl bei der Auswahlangabe als auch bei der Konstruktionsangabe zu achten.</li> <li>– Abhängig von der Konstruktionsangabe können aus unterschiedlichen Namen, die in der Auswahlangabe ausgewählt werden, identische Namen gebildet werden: z.B. "A/*" wählt die Namen "A1" und "A2" aus; die Konstruktionsangabe "B/*" erzeugt für beide Namen denselben neuen Namen "B". Um dies zu vermeiden, sollten in der Konstruktionsangabe alle Platzhalter der Auswahlangabe mindestens einmal verwendet werden.</li> <li>– Wird die Konstruktionsangabe mit einem Punkt abgeschlossen, so muß auch die Auswahlzeichenfolge mit einem Punkt enden. Die Zeichenfolge, die durch den Punkt am Ende der Auswahlzeichenfolge ausgewählt wird, kann in der Konstruktionsangabe nicht über den globalen Index angegeben werden.</li> </ul>

Tabelle 3: Zusätze zu Datentypen (Teil 5 von 7)

Zusatz	Bedeutung			
with-wild-constr (Forts.)	Beispiele:			
	Auswahlmuster	Auswahl	Konstruktionsmuster	neuer Name
	A//*	AB1 AB2 A.B.C	D<3><2>	D1 D2 D.CB
	C.<A:C>/<D,F>	C.AAD C.ABD C.BAF C.BBF	G.<1>.<3>.XY<2>	G.A.D.XYA G.A.D.XYB G.B.F.XYA G.B.F.XYB
	C.<A:C>/<D,F>	C.AAD C.ABD C.BAF C.BBF	G.<1>.<2>.XY<2>	G.A.A.XYA G.A.B.XYB G.B.A.XYA G.B.B.XYB
	A//B	ACDB ACEB AC.B A.CB	G/XY/	GCXYD GCXYE GCXY. <sup>1)</sup> G.XYC
	1) Punkt am Ende des Namens kann Namenskonvention widersprechen (z.B bei vollqualifizierten Dateinamen)			
without	Schränkt die Angabemöglichkeiten für einen Datentyp ein.			
-cat	Die Angabe einer Katalogkennung ist nicht erlaubt.			
-corr	Eingabeformat: [[C]' ][V][m]m.na[' ] Angaben zum Datentyp product-version dürfen den Korrekturstand nicht enthalten.			
-gen	Die Angabe einer Dateigeneration oder Dateigenerationsgruppe ist nicht erlaubt.			
-man	Eingabeformat: [[C]' ][V][m]m.n[' ] Angaben zum Datentyp product-version dürfen weder Freigabe- noch Korrekturstand enthalten.			
-odd	Der Datentyp x-text erlaubt nur eine gerade Anzahl von Zeichen.			
-sep	Beim Datentyp text ist die Angabe der folgenden Trennzeichen nicht erlaubt: ; = ( ) < > _ (also Strichpunkt, Gleichheitszeichen, runde Klammer auf und zu, Größerzeichen, Kleinerzeichen und Leerzeichen)			
-temp-file	Die Angabe einer temporären Datei ist nicht erlaubt (siehe #datei bzw. @datei bei filename).			

Tabelle 3: Zusätze zu Datentypen (Teil 6 von 7)

Zusatz	Bedeutung
without (Forts.) -user -vers -wild	<p>Die Angabe einer Benutzerkennung ist nicht erlaubt.</p> <p>Die Angabe der Version (siehe "datei(nr)") ist bei Banddateien nicht erlaubt.</p> <p>Die Datentype posix-filename bzw. posix-pathname dürfen keine Musterzeichen enthalten.</p>
mandatory -corr -man -quotes	<p>Bestimmte Angaben sind für einen Datentyp zwingend erforderlich.</p> <p>Eingabeformat: <code>[[C]' ][V][m]m.naso[' ]</code> Angaben zum Datentyp product-version müssen den Korrekturstand (und damit auch den Freigabestand) enthalten.</p> <p>Eingabeformat: <code>[[C]' ][V][m]m.na[so][' ]</code> Angaben zum Datentyp product-version müssen den Freigabestand enthalten. Die Angabe des Korrekturstands ist optional möglich, wenn dies nicht durch den Zusatz without-corr untersagt wird.</p> <p>Angaben zu den Datentypen posix-filename bzw. posix-pathname müssen in Hochkommata eingeschlossen werden.</p>

Tabelle 3: Zusätze zu Datentypen (Teil 7 von 7)

## 10.8 Kommando-Returncodes

SDF liefert dem Benutzer in einem Kommando-Returncode Informationen über die Analyse der Kommandoeingabe und die Kommandoausführung. Dieser Kommando-Returncode ist vergleichbar mit dem Returncode auf Programmebene. Er ermöglicht es dem Benutzer, auf bestimmte Fehlersituationen gezielt zu reagieren.

Der Kommando-Returncode besteht aus drei Teilen:

- Maincode, der einem Meldungsschlüssel entspricht, über den detaillierte Informationen mit dem Kommando /HELP-MSG-**INFORMATION** abgefragt werden können.
- Subcode1, der die aufgetretene Fehlersituation in eine Fehlerklasse einordnet, aus der abgeleitet werden kann, wie schwerwiegend ein Fehler ist. Der Wert von Subcode1 wird *dezimal* ausgegeben.
- Subcode2, der Zusatzinformationen (Wert ungleich Null) enthalten kann. Der Wert von Subcode2 wird *dezimal* ausgegeben. Für den Fehlerfall (also Subcode1 ungleich Null) ist die Verwendung von Subcode2 keinen Regeln unterworfen. Subcode2 kann außer Null die Werte 1 oder 2 annehmen, wenn kein Fehler aufgetreten ist. Subcode2 mit Wert 1 zeigt an, daß die angeforderte Leistung bereits vor Absetzen des Kommandos bestand. Subcode2 mit Wert 2 zeigt eine Sondersituation an und ist als Warnung einzustufen.

Der Kommando-Returncode kann nur mit SDF-P-Mitteln in S-Prozeduren und Dialogblöcken ausgewertet werden (siehe Handbuch „SDF-P“ [30]). Zur Auswertung siehe auch die Beschreibung des Kommandos IF-BLOCK-ERROR und der Builtin-Funktionen MAINCODE, SUBCODE1 und SUBCODE2 im Handbuch „Kommandos, Band 4“ [27].

### *Hinweis zur Darstellung*

Kommando-Returncodes werden in Tabellenform in der Reihenfolge Subcode2, Subcode1, Maincode und Bedeutungstext dargestellt. Ein Subcode2 mit dem Wert Null (d.h. es ist keine Zusatzinformation vorhanden) wird nicht in der Tabelle aufgeführt.

Ausführliche Informationen zum Kommando-Returncode finden Sie im Handbuch „Kommandos Band 1“ [27].

---

# Abkürzungen

AID	Advanced Interactive Debugger
ANSI	American National Standards Institute
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
BOOTP	BOOtstrap Protocol
BSD	Berkeley Software Distribution
CRTE	Common RunTime Environment
DCE	Distributed Computing Environment
DFS	Distributed File System
DHCP	Dynamic Host Configuration Protocol
DME	Distributed Management Environment
DMS	Data Management System
DNS	Domain Name Service
DSSM	Dynamic Subsystem Management
EBCDIC	Extended Binary Coded Decimal Interchange Code
EDT	EDiT
HSMS	Hierachical Storage Management System
HSMS-SV	Hierachical Storage Management System - SerVer

IEEE	Institute of Electrical and Electronics Engineers
ISAM	Indexed Sequential Access Method
ISO	International Organization for Standardization
LAN	Local Area Network
NFS	Network File System
OPS	Output Presentation Service
OSD	Open Systems Direction
OSF	Open Software Foundation
PAM	Physical Access Method
PLAM	Program Library Access Method
PLIB	POSIX Library
POSIX	Portable Open System Interface for UNIX
SAM	Sequential Access Method
SCI	Software Configuration Inventory
SIA	System Interfaces for Application
SNMP	Simple Network Management Protocol
SPOOL	Simultaneous Peripheral Operation On Line
SRPM	System Resources and Privileges Management
Sun	Sun Microsystems
TCP/IP	Transmission Control Protocol / Internet Protocol
TC-IP-AP	TCP/IP Application Programs
TCP-IP-SV	TCP/IP Services

TFTP	Trivial File Transfer Protocol
TIAM	Terminal Interactive Access Method
TLI	Transport Layer Interface
TOG	The Open Group
TPR	Task Privileged
TU	Task Unprivileged
TV	TransView
UDP/IP	User Datagram Protocol/Internet Protocol
UFS	UNIX File System
UNIX95	Synonym für XPG4.2
URL	Uniform Resource Locator
USL	UNIX System Laboratories
WAN	Wide Area Network
WWW	World Wide Web
X/Open	X/Open Company Ltd.
XPG4	X/Open Portability Guide Issue 4
XPG4.2	X/Open Portability Guide Issue 4, Version 2
XTI	X/Open Transport Interface



# Fachwörter

In diesem Verzeichnis sind die wichtigsten Begriffe dieses Handbuchs in alphabatischer Reihenfolge aufgeführt und erklärt. Bei Betriebssystem-spezifischen Fachwörtern ist angegeben, aus welcher Betriebssystem-Umgebung sie kommen (BS2000, POSIX oder UNIX).

Querverweise auf andere Fachwörter sind durch *Kursivdruck* gekennzeichnet.

## Abrechnungsnummer

account number

BS2000: Bezeichnet ein Abrechnungskonto für die zugehörige *Benutzerkennung*. Mehreren Benutzerkennungen kann dieselbe Abrechnungsnummer zugewiesen werden. Eine Benutzerkennung kann über maximal 60 Abrechnungsnummern verfügen. Die Abrechnungsnummer wird bei SET-LOGON-PARAMETERS und ENTER-JOB ausgewertet.

## Absoluter Pfadname

absolute pathname

POSIX/UNIX: *Pfadname* für eine *Datei* oder ein *Dateiverzeichnis*, der beim *Root-Verzeichnis* (/) des *POSIX-Dateisystems* beginnt und durch alle übergeordneten Dateiverzeichnisse führt. Jede Datei und jedes Dateiverzeichnis hat einen eindeutigen absoluten Pfadnamen.

## Aktuelles Dateiverzeichnis

current directory

POSIX/UNIX: *Dateiverzeichnis*, in dem der *Benutzer* gerade arbeitet; es kann mit dem POSIX-Kommando pwd angezeigt werden. Im aktuellen Dateiverzeichnis kann der Benutzer auf sämtliche *Dateien* und *Unterverzeichnisse* direkt zugreifen.

## Application Programming Interface (API)

Schnittstelle zwischen den Anwendungen und den von diesen benutzten System-/Subsystemfunktionen.

siehe auch *Distributed Computing Environment*

## ASCII

Abkürzung für **American Standard Code for Information Interchange**. Standardisierter Code zur Umwandlung von Groß- und Kleinbuchstaben, Ziffern, Sonder- und Steuerzeichen in digitale Ziffern, die im Rechner verarbeitet werden können. *UNIX* und *SINIX* arbeiten mit dem ASCII-Code. POSIX kann ASCII-Daten nach der Konvertierung in das EBCDIC-Format bearbeiten.

## Authentisierung

authentication

BS2000: Überprüfung der Angaben eines *Benutzers* beim Systemzugang. Die Benutzerattribute „*Benutzerkennung*“ und „*Kennwort*“ werden gegen die Einträge im *Benutzerkennungs-Katalog* geprüft.

## Behälterdatei

container file

POSIX: BS2000-PAM-Datei, in der ein *POSIX-Dateisystem* abgelegt ist. Eine Behälterdatei wird auf einem Pubset abgelegt. Behälterdateien und andere BS2000-Dateien dürfen auf demselben Pubset liegen.

## Benutzer

user

BS2000: Repräsentant einer *Benutzerkennung*. Der Begriff Benutzer ist ein Synonym für Personen, Anwendungen, Verfahren etc., die über eine Benutzerkennung Zugang zum *Betriebssystem* erhalten können.

## Benutzerattribute

user attributes

BS2000/POSIX: Alle Merkmale einer *Benutzerkennung*, die im *Benutzerkennungs-Katalog* hinterlegt sind.

## Benutzergruppe

user group

BS2000: Zusammenfassung einzelner *Benutzer* unter einem Namen.

## Benutzerkatalog

user catalog

siehe *Benutzerkennungs-Katalog*

**Benutzerkennung**

user identification, USER-ID

BS2000: Maximal acht Zeichen langer Name, der im *Benutzerkennungs-Katalog* eingetragen wird. Anhand der Benutzerkennung wird der *Benutzer* beim Systemzugang identifiziert. Alle *Dateien* und *Jobvariablen* werden unter einer Benutzerkennung eingerichtet. Die Namen der Dateien und Jobvariablen werden mit der Benutzerkennung im Dateikatalog hinterlegt.

**Benutzerkennungs-Katalog**

join file

BS2000: *Datei*, die die *Benutzerattribute* aller *Benutzerkennungen* eines *Pubsets* enthält.

**Benutzerkennungs-Kennwort**

password for user identification

Dient zur *Authentisierung* des *Benutzers* und gewährleistet den Zugangsschutz.

**Benutzerklasse Andere**

file other class

POSIX/UNIX: Eigenschaft einer *Datei*, die das *Zugriffsrecht* für einen *Prozeß* anzeigt, der mit der Benutzeridentifikation eines Prozesses verbunden ist. Ein Prozeß gehört zur Benutzerklasse Andere einer Datei, wenn er nicht der *Benutzerklasse Eigentümer* oder der *Benutzerklasse Gruppe* angehört.

**Benutzerklasse Eigentümer**

file owner class

POSIX/UNIX: Eigenschaft einer *Datei*, die das *Zugriffsrecht* für einen *Prozeß* anzeigt, der mit der Benutzer- und Gruppenidentifikation eines Prozesses verbunden ist. Ein Prozeß gehört zur Benutzerklasse Eigentümer einer Datei, wenn die effektive *Benutzernummer* des Prozesses zur Benutzernummer der Datei paßt.

**Benutzerklasse Gruppe**

file group class

POSIX/UNIX: Ein *Prozeß* gehört zur Benutzerklasse Gruppe einer Datei, wenn er nicht der *Benutzerklasse Eigentümer* angehört und die effektive *Gruppennummer* oder eine der zusätzlichen Gruppennummern des Prozesses zur Gruppennummer der Datei paßt.

**Benutzernummer**

user ID

POSIX/UNIX: Positive ganze Zahl, durch die ein Systembenutzer identifiziert wird.

## **Benutzerorganisation**

BS2000: Zusammenfassung von *Benutzerkennungen* zu *Benutzergruppen*. Dadurch wird die Nachbildung bestehender Organisationsformen ebenso gestattet wie die projektorientierte Zusammenfassung von *Benutzern*.

## **Benutzerrechte**

user privileges

BS2000: Alle an eine *Benutzerkennung* vergebenen und im *Benutzerkennungskatalog* hinterlegten Attribute, die Rechte darstellen.

## **Benutzerverwaltung**

user administration

siehe *Systemglobale Benutzerverwaltung*

## **Berkeley Software Distribution (BSD)**

UNIX der Universität von Kalifornien in Berkeley. Die letzte Version von BSD war 4.3. Sie wurde auf viele Rechnersysteme portiert und von zahlreichen Computerherstellern als ihr Standard-UNIX-Produkt übernommen.

## **Betriebssystem**

operating system

Gesamtheit aller Software- und Firmware-Programme, die, ohne auf einen bestimmten Anwendungsfall zugeschnitten zu sein, den Betrieb eines Computers ermöglichen. In der Regel wird das Betriebssystem vom Computerhersteller mitgeliefert.

## **Blockterminal**

block-mode terminal

Terminal, das keine zeichenweisen Ein- und Ausgabe-Operationen unterstützt.

## **Client**

client

Rechner, der von einem anderen Rechner (*Server*) Dienste in Anspruch nimmt. Ein Rechner kann gleichzeitig für bestimmte Funktionen als Client Dienste anfordern und für andere Rechner als Server Dienste zur Verfügung stellen.

**Client-Server-Architektur**

client server architecture

Systemarchitektur, in der Rechnerkapazitäten und Anwendungen auf Dienstnehmer (*Clients*) und Dienstanbieter (*Server*) verteilt werden.

Server-Funktionen werden überwiegend von Mainframe- und UNIX-Systemen ausgeübt, die dafür bestimmten Bedingungen genügen müssen.

Client-Funktionen haben vor allem PCs, Workstations und UNIX-Systeme.

Client- und Server-Systeme können beliebig kombiniert werden; jeder Client kann grundsätzlich auf jeden Server zugreifen.

**Dämonprozeß**

daemon

POSIX/UNIX: Systemprozeß, der permanent und meistens im Hintergrund abläuft; er führt allgemeine Aufgaben durch. Bekanntes Beispiel ist der Drucker-Dämon, der dafür sorgt, daß eine Datei ausgedruckt wird, während der Benutzer bereits wieder arbeitet.

**Datei**

file

UNIX: Eine Datei wird bei *UNIX* über einen Indexeintrag identifiziert. Dieser Eintrag enthält die Informationen, ob die Datei eine *normale Datei*, eine *Gerätedatei* oder ein *Dateiverzeichnis* ist. Eine normale Datei enthält Text, Daten, Programme oder sonstige Informationen. Eine Gerätedatei bezeichnet ein Gerät oder einen Teil eines Gerätes, wie zum Beispiel ein Laufwerk oder eine Festplattenpartition. Ein Dateiverzeichnis enthält andere Dateien.

BS2000: Sätze, die zueinander in Beziehung stehen, werden in einer benannten Einheit, der Datei, zusammengefaßt. Dateien sind beispielsweise: Konventionelle Ein-/Ausgabedaten von Programmen; Lademodule; Textinformation, die mit einem Dateiaufbereiter erstellt und verarbeitet wird.

**Dateisystem**

file system

POSIX/UNIX: Ansammlung von *Dateiverzeichnissen* und *Dateien* und bestimmter Attribute von Dateien.

**Dateiverzeichnis**

directory

POSIX/UNIX: Ein Dateiverzeichnis wird verwendet, um *Dateien* oder Dateiverzeichnisse zu gruppieren und zu organisieren.

**Distributed Computing**

siehe *Verteilte Verarbeitung*

### **Distributed Computing Environment (DCE)**

Von *OSF* genormte Umgebung für *verteilte Verarbeitung*. DCE ermöglicht Entwicklung und Ablauf von verteilten Anwendungen in einem heterogenen *Rechnernetz*.

DCE ist von *Betriebssystemen* und *Rechnernetzen* unabhängig. Es ermöglicht *Interoperabilität* und *Portabilität* zwischen heterogenen *Plattformen*.

### **Distributed Management Environment (DME)**

Produkt von OSF. Mit DME können heterogene Rechnerarchitekturen in einem *Netz* von einer einzigen Stelle aus verwaltet werden. DME läuft auf DCE ab.

### **EBCDIC**

Abkürzung für **Extended Binary Coded Decimal Interchange Code**.

EBCDI-Code ist ein auf 8 Bit erweiterter BCD-Code, der auf BS2000-Rechnern, TRANSDATA-Kommunikationsrechnern und Maschinen des Industrie-standards verwendet wird.

### **Einhängen von Dateisystemen**

mounting file systems

POSIX/UNIX: Mit dem Kommando *mount* können *Dateisysteme* in ein lokales Dateisystem eingehängt werden. Der *Einhängepunkt* muß vorher als *Dateiverzeichnis* definiert worden sein. Nach dem Einhängen des Dateisystems ist dieses Dateiverzeichnis nicht mehr sichtbar.

### **Einhängepunkt**

mount point

POSIX/UNIX: Name eines Dateiverzeichnisses, unter dem eine ferne Resource, z.B. ein Dateibaum, eingehängt ist.

### **Ethernet**

Standardverfahren zur Kopplung von zwei Rechnern; daraus entsteht ein *lokales Rechnernetz* (Local Area Network). Das Ethernet-Netz von Siemens Nixdorf heißt LAN1.

### **Ferner Rechner**

remote system

In einem lokalen *Netz* werden ferne und *lokale Rechner* unterschieden. Alle Rechner im Netz, an denen ein *Benutzer* nicht direkt arbeitet, sind für diesen Benutzer ferne Rechner.

**First-Start**

first start

BS2000: Beim First-Start des BS2000 werden Systemdateien neu eingerichtet. Das System vergibt eine Reihe von *Benutzerkennungen*, z.B. TSOS, SYSPRIV und SYSHSMS. Beim First-Start wird immer der *Benutzerkennungs-Katalog* angelegt.

**fork**

POSIX/UNIX: Systemaufruf, der einen *Prozeß* in zwei Prozesse teilt: den *Vater-* und den *Sohnprozeß*

**Gerätedatei**

special file

Eine auch als Gerätetreiber bezeichnete *Datei*, die als Schnittstelle zu einem Ein-/Ausgabegerät (z.B. Terminal, Plattenlaufwerk) benutzt wird.

**Gruppenmitglied**

group member

BS2000: *Benutzerkennung*, die einer *Benutzergruppe* zugeordnet ist. Der Gruppenverwalter kann einem Gruppenmitglied Ressourcen zuweisen.

**Gruppennummer**

group ID

Positive ganze Zahl zum Identifizieren einer Gruppe von Benutzern. Jeder Benutzer ist Mitglied mindestens einer Gruppe.

**Heterogenes Rechnernetz**

heterogeneous network

siehe *Offenes Rechnernetz*

**Hintergrundprozeß**

background process

*Prozeß*, der die Ressourcen des Rechners nicht vollständig ausschöpft, sondern die gleichzeitige Durchführung von weiteren Prozessen ermöglicht. Ein Hintergrundprozeß nutzt normalerweise die Zeitnischen aus, in denen der Prozessor sonst unbeschäftigt wäre.

**Home-Verzeichnis**

home directory

POSIX: *Dateiverzeichnis*, in das der *Benutzer* automatisch gelangt, wenn er mit POSIX verbunden wird.

### **Homogenes Rechnernetz**

homogeneous network

*Rechnernetz*, in dem die einzelnen Rechner dieselbe oder eine ähnliche Architektur haben.

### **Host**

host

Zentralrechner eines *Rechnernetzes*. Auf dem Host werden Programme durchgeführt, Dateien gespeichert sowie Ein- und Ausgaben gesteuert. In vielen Fällen enthält ein leistungsfähiges Rechnernetz mehrere Hosts.

### **Institute of Electrical and Electronics Engineers (IEEE)**

Organisation, die bedeutende Normen in der Computer- und Kommunikationsindustrie festgelegt hat, so z.B. die Norm IEEE1003 POSIX.

### **International Organization for Standardization (ISO)**

Internationale Normungsbehörde, die im Jahre 1946 gegründet wurde und inzwischen aus über 90 nationalen Normungsgremien besteht. ISO legt unter anderem auch Normen für die Software von *Rechnernetzen* fest.

### **Internet**

Weltweites *Rechnernetz* auf IP-Basis, das Tausende von verschiedenen Rechnernetzen miteinander verbindet und vom Network Information Center verwaltet wird. Über Internet können nur hardwareunabhängige Datenpakete verschickt werden.

### **Interoperabilität**

interoperability

Fähigkeit, Systeme von verschiedenen Computerherstellern miteinander verbinden zu können und sie zusammen arbeiten zu lassen, um Arbeitsanforderungen - wie z.B. verteilte Datenhaltung - erfüllen zu können. Dabei muß der *Benutzer* die Eigentümlichkeiten der einzelnen Systeme gar nicht oder nur wenig kennen.

### **ISO-Referenzmodell**

Rahmen für die Standardisierung der Kommunikation offener Systeme.

*ISO*, die internationale Organisation für Standardisierung, hat dieses Modell in dem internationalen Standard ISO 7498 beschrieben. Das ISO-Referenzmodell unterteilt die Funktionen, die für die Kommunikation von Systemen notwendig sind, in sieben logische Schichten. Diese Schichten haben jeweils klar definierte Schnittstellen zu den benachbarten Schichten und kommunizieren mit den jeweils entsprechenden Schichten auf dem Partner-Rechner über *Protokolle*.

**Jobvariable**

job variable

BS2000: Jobvariablen sind Speicherbereiche zum Austausch von Informationen zwischen Aufträgen (Jobs) untereinander sowie zwischen *Betriebssystem* und Aufträgen. Sie haben einen Namen und einen Inhalt (Wert). Der Inhalt kann zur Steuerung von Aufträgen und Programmen genutzt werden.

Der *Benutzer* kann Jobvariablen erzeugen, verändern, abfragen und löschen. Außerdem kann er das Betriebssystem anweisen, eine überwachende Jobvariable entsprechend zu setzen, wenn sich der Zustand eines Auftrags oder eines Programms ändert.

**Katalogkennung**

catalog identification, catid

BS2000: Kennzeichen eines *Pubsets*.

Die Katalogkennung besteht aus maximal 4 Zeichen, die immer in Doppelpunkte eingeschlossen werden müssen. Die Katalogkennung ist Bestandteil des *Pfadnamens* einer *Datei*.

**Kennwort**

password

Folge von Zeichen, die der *Benutzer* eingeben muß, um den Zugriff zu einer *Benutzerkennung*, einer *Datei*, einer *Jobvariablen*, einem Netzknoten oder einer Anwendung zu erhalten.

**Kompatibilität**

compatibility

Fähigkeit, Tasks ohne größere Änderungen in verschiedenen Systemumgebungen mit gleichem Ergebnis durchführen zu können.

**LAN1**

siehe *Ethernet*

**Local Area Network (LAN)**

Hardware-Konfiguration eines lokalen *Netzes*, in dem alle Datensichtgeräte und sonstigen Geräte in relativ geringem Abstand zueinander aufgestellt sind, z.B. innerhalb desselben Gebäudes. Die geringe Entfernung erlaubt einfachere Übertragungstechniken und damit höhere Geschwindigkeiten zu einem geringen Preis.

In der Bundesrepublik Deutschland ist die Größe eines LAN auf das Grundstück des Anwenders beschränkt. Ein LAN kann als privates Subnetz mit anderen *Rechnernetzen* verbunden und so Teil eines größeren Netzes sein, etwa eines WAN.

Synonyme: Lokales Rechnernetz, lokales Netz.

**Login-Verzeichnis**

login directory

siehe *Home-Verzeichnis*

**Lokales Dateisystem**

local file system

POSIX/UNIX:

**Lokales (Rechner-)Netz**

siehe *Local Area Network*

**Lokaler Rechner**

local system

Für einen *Benutzer* ist immer derjenige Rechner lokal, an dem er arbeitet. Alle anderen Rechner im *Rechnernetz* sind dann für ihn *ferne Rechner*.

**Mehrfach benutzbare Datei**

shareable file

BS2000: Eine *Datei*, die der *Benutzer* mit dem Operanden USER-ACCES= \*ALL-USERS katalogisiert hat. Dateien, die auf diese Weise als mehrfach benutzbar gekennzeichnet sind, können von allen Benutzern aufgerufen werden. Voraussetzung ist aber, daß der Benutzer die *Benutzerkennung* des Erstellers der Datei kennt und ggf. das *Kennwort* angeben kann, falls die Datei geschützt ist.

**Mini-POSIX-Dateisystem**

POSIX: NFS- oder DFS-Dateisystem, das mit BS2000/OSD V1.0 oder früheren BS2000-Versionen erstellt wurde. Es kann in *POSIX-Dateisysteme* migriert werden.

**Motif**

siehe *OSF/Motif*

**Network File System (NFS)**

BS2000: Softwareprodukt, mit dem verteilte Datenhaltung in einem heterogenen *Rechnernetz* möglich ist. Der *Benutzer* kann auf ferne *Dateien* so zugreifen, als ob sie an seinem *lokalen Rechner* vorhanden wären.

**Netz**

network

Komplexes Gebilde aus Leitungen und Steuerungseinrichtungen, das der Datenfernübertragung dient.

**Normale Datei**

regular file

*Datei*, die eine wahlfrei zugreifbare Folge von Bytes ohne jede weitere vom System festgelegte Struktur ist.

**Offenes Rechnernetz**

open network

*Rechnernetz*, in dem nach den Regeln von *ISO* kommuniziert wird. Durch festgelegte *Protokolle* können Rechner von verschiedenen Computerherstellern miteinander arbeiten.

**Open Software Foundation (OSF)**

Forschungs- und Entwicklungsgesellschaft, die sich zum Ziel gesetzt hat, Softwarelösungen bereitzustellen, mit der Rechner von vielen Computerherstellern in einer offenen Systemumgebung zusammenarbeiten können.

Zu den OSF-Produkten zählen *OSF/1*, *OSF/Motif*, DCE und DME.

Gründungsmitglieder von OSF im Jahre 1988 waren u.a. Siemens und Nixdorf.

**OSF/1**

Auf *UNIX* basierendes *Betriebssystem*, das von OSF entwickelt wurde.

**OSF/Motif**

Grafische Benutzeroberfläche der *UNIX*-Industrienorm, die ursprünglich bei OSF entwickelt wurde. OSF/Motif basiert auf dem X Windows System.

**Pfadname**

pathname

POSIX/UNIX: Jede *Datei* und jedes *Dateiverzeichnis* besitzt einen eindeutigen Pfadnamen. Der Pfadname gibt die Position der Datei bzw. des Dateiverzeichnisses innerhalb des *Dateisystems* an und zeigt, wie darauf zugegriffen werden kann. Der Pfadname besteht aus den Namen aller darüberliegenden Dateiverzeichnisse, ausgehend von der Spitze des Dateisystems, und dem eigentlichen Namen der Datei oder des Dateiverzeichnisses. Die Namen der Dateiverzeichnisse werden jeweils durch einen Schrägstrich (Slash) voneinander getrennt. (Beispiel: c:verzeich1/verzeich2/protokoll)

Es wird zwischen *absoluten* und *relativen Pfadnamen* unterschieden.

BS2000: Jede im BS2000 katalogisierte Datei ist ebenfalls durch einen Pfadnamen eindeutig identifizierbar. Der Pfadname setzt sich zusammen aus der *Katalogkennung* (catid), der *Benutzerkennung* (userid) und einem vom Benutzer vergebenen vollqualifizierten Dateinamen:

:catid:\$userid.dateiname

### Pipe pipe

POSIX/UNIX: Verkettung zweier *POSIX/UNIX*-Kommandos. Eine Pipe bewirkt, daß die Ausgabe eines Programms die Eingabe des nächsten Programms wird, so daß die Programme nacheinander ausgeführt werden. Eine Pipe wird erzeugt, indem nach dem ersten Kommando das Pipe-Symbol | angegeben wird. Die Ausgabe des Prozesses links vom Pipe-Symbol wird an den *Prozeß* rechts vom Pipe-Symbol geleitet.

### Plattform

Betriebssystemumgebung, in der ein Programm abläuft.

### Portabilität

#### portability

Fähigkeit eines Programms, unverändert auf unterschiedlichen *Betriebssystemen* ablaufen zu können. Sie wird durch die Verwendung standardisierter, offener Programmschnittstellen erreicht, die auf einer Vielzahl von *Plattformen* angeboten werden.

### Portable Open System Interface for UNIX (POSIX)

Schnittstellennormen für offene Systeme, die von der IEEE festgesetzt wurden und auf UNIX basieren. POSIX enthält Normen für ein breites Spektrum von Betriebssystemkomponenten, angefangen von der Programmiersprache C bis zur Systemverwaltung. POSIX-Arbeitsgebiete sind unter anderem:

- 1003.00 Guide to POSIX Open System Environment
- 1003.01 System Application Program Interface (API)
- 1003.02 Shell and Utilities
- 1003.07 System Administration

### Portierbarkeit

siehe *Portabilität*

### POSIX.1-1988

IEEE Std 1003.1-1988, Standard für Informationstechnologie - POSIX - Teil 1: System Application Program Interface (API)

**POSIX-Dateisystem**

POSIX file system

BS2000/POSIX: *Dateisystem* im BS2000 mit der Struktur eines UNIX-Dateisystems (UFS). Es kann aus mehreren Dateisystemen bestehen. Es ist hierarchisch aufgebaut und besteht aus *Dateiverzeichnissen* und *Dateien* (POSIX-Dateien). An der Spitze der Hierarchie steht das Dateiverzeichnis *root*, das durch einen Schrägstrich (/) gekennzeichnet ist. Von hier aus setzt sich die Verzeichnisstruktur weiter nach unten fort. Von Dateiverzeichnissen aus kann in weitere Dateiverzeichnisse oder in Dateien verzweigt werden. Von einer Datei aus ist keine Verzweigung mehr möglich. Jede Datei eines Dateisystems ist über genau einen absoluten Pfad erreichbar.

Der Unterschied zwischen einem POSIX- und einem UNIX-Dateisystem besteht im Ablageort: Bei einem UNIX-Dateisystem ist der Ablageort ein physikalisches Gerät, bei einem POSIX-Dateisystem eine PAM-*Behälterdatei*.

**POSIX-Shell**

POSIX shell

BS2000/POSIX: Portiertes *SINIX*-Systemprogramm, das die Kommunikation zwischen dem *Benutzer* und dem System übernimmt. Die POSIX-Shell ist ein Kommando-Interpreter. Sie übersetzt die eingegebenen POSIX-Kommandos in eine Sprache, die das System erkennt.

Wenn beim Benutzerattribut „Programm“ die POSIX-Shell eingetragen ist, wird die POSIX-Shell gestartet, sobald sich der Benutzer durch Remote Login an POSIX angeschlossen hat.

**POSIX-Verwalter**

POSIX administrator

Inhaber des Privilegs **POSIX-ADMINISTRATION**. Dieses Privileg ist automatisch an die Systemkennung **SYSROOT** geknüpft und kann **SYSROOT** nicht entzogen werden. Anderen *Benutzerkennungen* kann der Sicherheitsbeauftragte dieses Privileg auch verleihen und entziehen.

**Protokoll**

protocol

Regeln für den Datenaustausch zwischen zwei Rechnern, die die Art der Verbindung, das Datenformat sowie die Abfolge der Daten bestimmen.

**Prozeß**

process

POSIX/UNIX: Adreßraum, in dem ein einzelner Programmcode ausgeführt wird, sowie die dafür benötigten Betriebsmittel des Systems.

Ein Prozeß wird von einem anderen Prozeß durch den Aufruf der Funktion *fork* erzeugt. Der Prozeß, der *fork* aufruft, heißt Vaterprozeß; der durch *fork* erzeugte Prozeß heißt Sohnprozeß.

## **Public Volume Set**

BS2000: Satz gemeinschaftlich gekennzeichneter Platten. MPVS-Systeme arbeiten mit mehreren voneinander unabhängigen Pubsets.

## **Pubset**

BS2000: Kurzform für *Public Volume Set*

## **Rechnernetz**

network

Zusammenschluß mehrerer Rechner über eine physikalische Verbindung mit dem Ziel, einen gleichberechtigten Datenaustausch zwischen diesen Rechnern zu ermöglichen. Es gibt lokale (LAN) und weite (WAN) Rechnernetze, *heterogene* und *offene Rechnernetze*.

## **Rechnernetz, offenes**

siehe *Offenes Rechnernetz*

## **Relativer Pfadname**

relative pathname

POSIX: Pfadname für eine *Datei* oder ein *Dateiverzeichnis*, der von der Position des *aktuellen Dateiverzeichnisses* innerhalb des *Dateisystems* ausgeht. Relative Pfadnamen beginnen nicht mit einem Schrägstrich (/).

## **Reliant UNIX**

Nachfolger von *SINIX*, das mit der Version 5.43 als Konsequenz aus der Zusammenführung der UNIX-Versionen von Siemens Nixdorf und Pyramid Technology in Reliant UNIX umbenannt wurde. Der Name Reliant UNIX steht für die hohen Anforderungen an Zuverlässigkeit und Verfügbarkeit, die dieses standardisierte Betriebssystem erfüllt – im kommerziellen wie im technischen Einsatz – wobei alle bewährten *SINIX*-Eigenschaften in Reliant UNIX erhalten bleiben.

## **root**

root

POSIX/UNIX: Benutzername (Systemverwalter-Kennung) mit den meisten Privilegien.

## **Root-Berechtigung**

Kennung, der die *Benutzernummer* 0 und die *Gruppennummer* 0 zugeteilt ist. Die Systemkennung SYSROOT hat standardmäßig die Root-Berechtigung.

**Root-Verzeichnis**

root directory

POSIX/UNIX: Hauptdateiverzeichnis in einem hierarchisch strukturierten Dateisystem, von dem alle anderen Dateiverzeichnisse abzweigen.

**Schutzattribute**

security attributes

BS2000: Sicherheitsrelevante Eigenschaften eines Objekts (*Datei, Jobvariable etc.*), die die Art und Möglichkeit des Zugriffs auf dieses Objekt festlegen.

Für Dateien gibt es beispielsweise folgende Schutzattribute: ACCESS/USER-ACCESS, SERVICE-bit, AUDIT-Attribut, RDPASS, WRPASS, EXPASS, RETPD, BACL, ACL und GUARD.

**Schutzbits**

permission bits

POSIX/UNIX: Information über Lese-, Schreib- oder Ausführungsrecht einer Datei. Die Bits sind in drei Abschnitte eingeteilt: Eigentümer, Gruppe und andere Benutzer.

**Semaphor**

semaphore

POSIX/UNIX:

**Server**

server

Rechner, der anderen Rechnern (*Clients*) Dienste zur Verfügung stellt.

**shutdown**

siehe *Systembeendigung*

**SINIX**

Siemens Nixdorf Version des *UNIX-Betriebssystems* der Firma AT&T. SINIX erfüllt alle *X/Open*-Normen und garantiert durch viele weitere Weltstandards konsequente Offenheit.

**Sitzung**

session

Vorgänge/Aktivitäten, die zwischen der *Systemeinleitung* und der *Systembeendigung* stattfinden.

**Sockets**

Schnittstelle für Netzzugriffe über TCP/IP.

### **Sohnprozeß**

child process

siehe *fork*

### **Systembeendigung**

shutdown

BS2000: Vorgang der geordneten Systembeendigung einschließlich des Sicherns spezieller Systemdateien.

### **Systemeinleitung**

startup

BS2000: Laden der *Betriebssystem*-Software des BS2000. Es gibt vier Varianten:

- AUTOMATIC-STARTUP
- DIALOG-STARTUP
- FAST-STARTUP
- QUICK-STARTUP

Diese Varianten unterscheiden sich durch unterschiedlichen Automatisierungsgrad und unterschiedlichen Rückbezug auf die letzte Systemeinleitung.

### **Systemglobale Benutzerverwaltung**

user administration

BS2000: Umfaßt die Verwaltung von *Benutzerkennungen* und *Benutzergruppen* bezüglich Ressourcen und Benutzerrechten sowie das Neueinrichten, Modifizieren und Löschen von Benutzerkennungen und Benutzergruppen.

### **Systemglobale Privilegien**

user privileges

BS2000: Alle Rechte, die mit dem Kommando /SET-PRIVILEGE vergeben werden können, sowie das Recht des Sicherheitsbeauftragten und der Systemkennung TSOS.

### **Systemkern**

kernel

POSIX/UNIX: Code des *POSIX-/UNIX-Betriebssystems*.

### **Systemverwalterrechte**

siehe *Systemglobale Privilegien*

### **TCP/IP**

Netzprotokoll der Internet-Architektur.

**UNIX File System (UFS)**

UNIX-Versionen <= IV:

Dateiverwaltungs-Komponente des *UNIX-Systemkerns*.

System V:

*Dateisystem*-Variante des Virtual File System für die Verwaltung lokaler Dateisysteme.

**UNIX-Betriebssystem**

Ein im Dialogbetrieb arbeitendes *Betriebssystem*, das 1969 von Bell Laboratories entwickelt wurde. Da nur ein zentraler *Systemkern* von UNIX hardwareabhängig ist, wird UNIX auf vielen unterschiedlichen Systemen verschiedener Computerhersteller eingesetzt. Das UNIX von Siemens Nixdorf heißt *SINIX*.

**UNIX95**

Synonym für *XPG4.2*

**Unterverzeichnis**

subdirectory

POSIX/UNIX: *Dateiverzeichnis*, über dem ein weiteres Dateiverzeichnis liegt.

**Vaterprozeß**

parent process

siehe *fork*

**Verteilte Verarbeitung**

Distributed Computing

Die Anwendungen und meistens auch die Daten und Ressourcen sind zwischen den einzelnen Rechnern eines *Netzes* aufgeteilt.

**Verzeichnis**

siehe *Dateiverzeichnis*

**Vordergrundprozeß**

foreground process

*Prozeß*, der das Terminal vollständig in Beschlag nimmt, so daß es von anderen Prozessen nicht genutzt werden kann.

**Wide Area Network (WAN)**

*Rechnernetz*, das nicht auf ein räumlich begrenztes Gebiet beschränkt ist.

### **X/Open**

Unabhängige, weltweite Organisation für offene Systeme, die von fast allen großen Computerherstellern, Benutzerverbänden und Softwaregesellschaften unterstützt wird. Ziel von X/Open ist die Implementierung offener Systeme, damit die *Benutzer* ihre Rechner besser einsetzen können. X/Open wirkt bei einer Vielzahl von internationalen Normierungsgremien mit.

### **XPG4**

*X/Open* Portability Guide, Issue 4.  
Sammlung der X/Open-Schnittstellenstandards.

### **XPG4.2**

*X/Open* Portability Guide, Issue 4, Version 2.  
Erweiterung von *XPG4*.

### **Zugriffsrecht**

access permission

POSIX/UNIX: Eigenschaft einer *Datei*, die den Zugriff auf diese Datei steuert. Zugriffsrechte werden getrennt vergeben an den Eigentümer (siehe *Benutzerklasse Eigentümer*), die Gruppe des Eigentümers (siehe *Benutzerklasse Gruppe*) und alle anderen Benutzer (siehe *Benutzerklasse Andere*). Es gibt drei grundlegende Zugriffsrechte: Lese-, Schreib- und Aufführungsrecht.

---

# Literatur

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

- [1] **POSIX im BS2000/OSD**  
**Die Öffnung zur UNIX-Welt**  
Allgemeine Beschreibung
- [2] **POSIX V1.1A (BS2000/OSD)**  
**Kommandos**  
Benutzerhandbuch

*Zielgruppe*

Das Handbuch wendet sich an alle Benutzer der POSIX-Shell.

*Inhalt*

Dieses Handbuch ist ein Nachschlagewerk. Es beschreibt das Arbeiten mit der POSIX-Shell sowie die Kommandos der POSIX-Shell in alphabetischer Reihenfolge.

- [3] **POSIX V1.1A (BS2000/OSD)**  
**SOCKETS/XTI für POSIX**  
Benutzerhandbuch

*Zielgruppe*

C- und C++-Programmierer, die mit SOCKETS- bzw. XTI-Funktionen Kommunikationsanwendungen auf der Basis der POSIX-Schnittstelle entwickeln.

*Inhalt*

- Einführung in SOCKETS(POSIX)
- Benutzerfunktionen von SOCKETS(POSIX)
- Einführung in XTI(POSIX)
- XTI-Trace
- Bibliotheksfunktionen von XTI(POSIX)
- Übersetzen und Binden von Kommunikationsanwendungen
- Konfiguration und Konfigurationsdateien, BCAM-Abhängigkeiten
- Einschränkungen zur Kompatibilität

[4] **C/C++ V2.2A (BS2000/OSD)**

**POSIX-Kommandos des C- und des C++-Compilers**  
Benutzerhandbuch

*Zielgruppe*

C- und C++-Anwender im BS2000/OSD.

*Inhalt*

- Einführung in die C/C++-Programmentwicklung in POSIX-Shell-Umgebung.
- Übersetzen und Binden von C- und C++-Programmen mit den POSIX-Kommandos *cc*, *c89* und *CC*.
- Steuern des globalen C- und C++-Listengenerators mit dem POSIX-Kommando *ccxref*.

[5] **C/C++ V3.0 (BS2000/OSD)**

**POSIX-Kommandos des C/C++-Compilers**

Benutzerhandbuch

*Zielgruppe*

C- und C++-Anwender im BS2000/OSD.

*Inhalt*

- Einführung in die C/C++-Programmentwicklung in POSIX-Shell-Umgebung.
- Übersetzen und Binden von C- und C++-Programmen mit den POSIX-Kommandos *cc*, *c89* und *CC*.
- Steuern des globalen C/C++-Listengenerators mit dem POSIX-Kommando *cclistgen*.

[6] **C-Bibliotheksfunktionen (BS2000/OSD)**

**für POSIX-Anwendungen**

Referenzhandbuch

*Zielgruppe*

C- und C++-Programmierer

*Inhalt*

Das Handbuch dokumentiert die XPG4-konforme C-Programmierschnittstelle, die vom POSIX-Subsystem im BS2000 unterstützt wird. Mit dieser Programmierschnittstelle kann sowohl auf das POSIX-Dateisystem als auch auf BS2000-Dateien zugegriffen werden. Zusätzlich enthält die Programmierschnittstelle Erweiterungen, die die Kompatibilität mit der bisherigen C-Bibliothek gewährleisten.

- [7] **CRTE (BS2000/OSD)**  
**Common RunTime Environment**  
Benutzerhandbuch  
*Zielgruppe*  
Programmierer und Systemverwalter im BS2000  
*Inhalt*  
Beschreibung der gemeinsamen Laufzeitumgebung für COBOL85-, C-, und C++-Objekte sowie für "Fremdsprachenmix":  
– Komponenten des CRTE  
– Programmkomunikationsschnittstelle ILCS  
– Bindebeispiele
- [8] **DCE (BS2000)**  
**POSIX-Programmschnittstelle**  
Benutzerhandbuch  
*Zielgruppe*  
Programmierer von DCE-Anwendungen  
*Inhalt*  
Beschreibung der C-Bibliotheksfunktionen mit POSIX-Funktionalität, die zum Programmieren von DCE-Anwendungen in BS2000/OSD benötigt werden.
- [9] **DCE (BS2000)**  
**Grundlagen, Einsatz und Administration**  
Benutzerhandbuch  
*Zielgruppe*  
DCE-Anwender und DCE-Administratoren  
*Inhalt*  
Überblick über DCE im BS2000, Beschreibung der BS2000-spezifischen Aufgaben für DCE-Anwender und für DCE-Administratoren.
- [10] **DCE (BS2000)**  
**Grundlagen, Einsatz und Administration**  
Referenzhandbuch  
*Zielgruppe*  
DCE-Anwender, DCE-Administratoren und Programmierer von DCE-Anwendungen  
*Inhalt*  
Enthält die SDF-Kommandos und die Administrationsprogramme, die für DCE im BS2000 zur Verfügung gestellt werden, die Meldungen und ein Fachwortverzeichnis.

- [11] **DCE (BS2000)**  
**Programmieren von Anwendungen**  
Benutzerhandbuch
- Zielgruppe*  
DCE-Programmierer
- Inhalt*  
Darstellung der BS2000-spezifischen Ausprägungen für die Programmierung, Anwendungsbeispiel, Programmschnittstellen, UUID-Generator, IDL-Compiler sowie Übersetzen, Binden, Laden und Starten von DCE-Anwendungen.
- [12] **NFS V1.2A (BS2000/OSD)**  
**Network File System**  
Benutzerhandbuch
- 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.
- [13] **SECOS (BS2000/OSD)**  
**Security Control System**  
Benutzerhandbuch
- Zielgruppe*
- BS2000-Systemverwalter
  - BS2000-Anwender, die den erweiterten Zugriffsschutz für Dateien nutzen
- Inhalt*  
Leistung und Anwendung der vier Funktionseinheiten:
- SRPM (Privilegien und Betriebsmittel verwalten)
  - FACS (erweiterter Zugriffsschutz für Dateien)
  - GUARDS (Zugriffsbedingungsverwaltung und -auswertung für Objekte)
  - SAT (Protokollierung und Auswertung sicherheitsrelevanter Daten, Ereignisüberwachung mit Alarmfunktion).

[14] **BS2000/OSD-BC V3.0****Bindelader-Starter**

Benutzerhandbuch

*Zielgruppe*

Das Handbuch wendet sich an Software-Entwickler und geübte BS2000/OSD-Benutzer.

*Inhalt*

Es beschreibt die Funktionen, die Unterprogrammschnittstelle, die XS-Unterstützung und den Aufruf des Bindeladers DBL.

Daran anschließend sind die Kommandos zum Aufruf des Laders ELDE und die Migration vom DLL zum DBL beschrieben.

[15] **C++ V2.2A (BS2000/OSD)****C++-Compiler**

Benutzerhandbuch

*Zielgruppe*

C- und C++-Anwender im BS2000.

*Inhalt*

- Beschreibung aller Tätigkeiten zum Erzeugen von ablauffähigen C- und C++-Programmen: Übersetzen, Binden, Laden, Testen;
- Programmierhinweise und weitergehende Informationen zu: Optimierung, Programmablaufsteuerung, Funktions- und Sprachverknüpfung, C- und C++-Sprachumfang des Compilers.

[16] **C/C++ V3.0 (BS2000/OSD)****C/C++-Compiler**

Benutzerhandbuch

*Zielgruppe*

C- und C++-Anwender im BS2000/OSD.

*Inhalt*

- Beschreibung aller Tätigkeiten zum Erzeugen von ablauffähigen C- und C++-Programmen: Übersetzen, Binden, Laden, Testen;
- Programmierhinweise und weitergehende Informationen zu: Optimierung, Programmablaufsteuerung, Funktions- und Sprachverknüpfung, C- und C++-Sprachunterstützung des Compilers.

- [17] **EDT V16.6 (BS2000/OSD)**

**Anweisungen**

Benutzerhandbuch

*Zielgruppe*

Das Handbuch wendet sich an EDT-Einsteiger und EDT-Anwender.

*Inhalt*

Das Handbuch beschreibt das Bearbeiten von SAM- und ISAM-Dateien, Elementen aus Programm-Bibliotheken und POSIX-Dateien. Es enthält weiter eine Beschreibung der Arbeitsmodi, Kurzanweisungen, EDT-Prozeduren und Anweisungen des EDT.

- [18] **BS2000/OSD-BC V3.0**

**Einführung in die Systembetreuung**

Benutzerhandbuch

*Zielgruppe*

Das Handbuch wendet sich an die Systembetreuung und das Operating des Betriebssystems BS2000/OSD.

*Inhalt*

Es sind u.a. folgende Themen zur Verwaltung und Überwachung des BS2000/OSD-Grundausbaus enthalten: Systemeinleitung, Parameterservice, Job- und Tasksteuerung, Speicher-, Geräte-, Benutzer-, Datei- und Pubset-Verwaltung, Privilegienvergabe, Accounting und Operatorfunktionen.

Wesentliche Neuheiten in der OSD-BC V3.0 sind das SMS-Konzept und HIPLEX MSCF.

- [19] **openFT (BS2000) V6.0**

**Enterprise File Transfer in der offenen Welt**

Benutzerhandbuch

*Zielgruppe*

Das Handbuch richtet sich an Benutzer, die mit *openFT* Dateien übertragen oder verwalten möchten.

*Inhalt*

Das Benutzerhandbuch stellt die Leistungen von *openFT* vor. Die Beschreibung beinhaltet auch die optionalen Komponenten *openFT-AC* für den Zugangs- und Zugriffsschutz und *openFT-OS* zur Unterstützung der FTAM-Funktionalität. Die Kommandoschnittstelle und Meldungen werden ausführlich dargestellt.

[20] ***openFT (UNIX) V6.0***

**Enterprise File Transfer in der offenen Welt**

Benutzerhandbuch

*Zielgruppe*

Das Handbuch wendet sich an Benutzer und Programmierer, die mit *openFT (UNIX)* Dateien übertragen oder Dateien verwalten möchten.

*Inhalt*

Das Handbuch beschreibt die Leistungen von *openFT (UNIX)* einschließlich dem Einsatz der FTAC-Funktionalität und der Unterstützung von FTAM. Es enthält die Kommandoschnittstellen, die Programmschnittstelle und die Meldungen.

[21] ***openFT (UNIX) V6.0***

**Enterprise File Transfer in der offenen Welt**

Installation und Administration

Systemverwalterhandbuch

*Zielgruppe*

Das Handbuch wendet sich an den Verwalter von *openFT (UNIX)*.

*Inhalt*

Das Handbuch beschreibt die Installation und Konfiguration von *openFT (UNIX)* und enthält die Kommandoschnittstelle zur Verwaltung von *openFT (UNIX)*. Die Verwaltung über SNMP wird ebenfalls beschrieben.

[22] **HSMS / HSMS-SV (BS2000/OSD)**

**Hierarchisches Speicher Management System**

Benutzerhandbuch

*Zielgruppe*

- BS2000/OSD-Benutzer
- BS2000/OSD-Systembetreuer
- HSMS-Verwalter

*Inhalt*

- Beschreibung der Funktionen Datensicherung, Langzeitarchivierung, Verdrängung und Datentransfer
- Verwaltung, Aufruf, Ablauf und Installation von HSMS
- Beschreibung der Anweisungen und Meldungen

[23] **IMON (BS2000/OSD)**

### **Installationsmonitor**

Benutzerhandbuch

#### *Zielgruppe*

Das Handbuch wendet sich an den BS2000/OSD-Systemverwalter.

#### *Inhalt*

Das Handbuch beschreibt die Arbeitsweise von IMON und seinen vier Komponenten IMON-BAS, IMON-GPN, IMON-XT und IMON-SIC. Sie erfahren wie BS2000-Software mit IMON und SOLIS2 installiert und registriert wird.

[24] **JV (BS2000/OSD)**

### **Jobvariablen**

Benutzerhandbuch

#### *Zielgruppe*

Das Handbuch wendet sich sowohl an den nichtprivilegierten Anwender als auch an die Systembetreuung.

#### *Inhalt*

Es beschreibt die Verwaltung und die verschiedenen Einsatzmöglichkeiten von Jobvariablen. Die Kommandobeschreibungen sind getrennt nach den Funktionsbereichen der JVs aufgeführt. Die Makroaufrufe sind in einem eigenen Kapitel beschrieben.

[25] **SDF (BS2000/OSD)**

### **SDF-Verwaltung**

Benutzerhandbuch

#### *Zielgruppe*

Das Handbuch wendet sich an die Systemverwaltung und an erfahrene BS2000-Benutzer

#### *Inhalt*

Es beschreibt, wie SDF mit Hilfe von SDF-Kommandos und den Dienstprogrammen SDF-I, SDF-U und SDF-PAR installiert und verwaltet wird. Die Anweisungen von SDF-I, SDF-U und SDF-PAR sind vollständig beschrieben.

[26] **SDF-A (BS2000/OSD)**

Benutzerhandbuch

#### *Zielgruppe*

Das Handbuch wendet sich an erfahrene BS2000-Benutzer und an die Systemverwaltung.

#### *Inhalt*

Es beschreibt, wie Syntaxdateien bearbeitet werden und erklärt die SDF-A-Funktionen anhand von Beispielen. Die SDF-A-Anweisungen sind in alphabetischer Reihenfolge aufgeführt.

Das Dienstprogramm SDF-SIM ist ebenfalls im Handbuch beschrieben.

[27] **BS2000/OSD-BC V3.0**  
**Kommandos Band 1, A-L**  
Benutzerhandbuch

*Zielgruppe*

Das Handbuch wendet sich sowohl an den nichtprivilegierten Anwender als auch an die Systembetreuung.

*Inhalt*

Es enthält die Kommandos ADD-... bis LOGOFF (BS2000/OSD-Grundausbau und ausgewählte Produkte) mit der Funktionalität für alle Privilegien. Die Einleitung gibt Hinweise zur Kommandoeingabe.

**Kommandos Band 2, M-SG**

Benutzerhandbuch

*Zielgruppe*

Das Handbuch wendet sich sowohl an den nichtprivilegierten Anwender als auch an die Systembetreuung.

*Inhalt*

Es enthält die Kommandos MODIFY... bis SET... (BS2000/OSD-Grundausbau und ausgewählte Produkte) mit der Funktionalität für alle Privilegien.

**Kommandos Band 3, SH-Z**

Benutzerhandbuch

*Zielgruppe*

Das Handbuch wendet sich sowohl an den nichtprivilegierten Anwender als auch an die Systembetreuung.

*Inhalt*

Das Handbuch enthält die Kommandos SHOW-... bis WRITE-... (BS2000/OSD-Grundausbau und ausgewählte Produkte) mit der Funktionalität für alle Privilegien. Mit SDF-P können Anwender bei SHOW-Kommandos die Ausgabe in strukturierte S-Variablen, nutzen die im Band 4 beschrieben ist.

**Kommandos Band 4, Ausgabe in S-Variablen und SDF-P-BASYS**

Benutzerhandbuch

*Zielgruppe*

Das Handbuch wendet sich an Programmierer und Anwender, die Prozeduren erstellen.

*Inhalt*

Das Handbuch enthält die tabellarische Darstellung aller S-Variablen, die von den SHOW-Kommandos bei einer strukturierten Ausgabe mit Werten versorgt werden.

Weitere Kapitel:

- Einführung in das Arbeiten mit S-Variablen
- SDF-P-BASYS V2.1A

- [28] **BS2000/OSD-BC V3.0**  
**Verwaltung von Subsystemen (DSSM/SSCM)**  
Benutzerhandbuch
- Zielgruppe*  
Das Handbuch wendet sich an die Systembetreuung und die Softwareberatung des BS2000/OSD.
- Inhalt*  
Es werden das Subsystemkonzept des BS2000, die Dynamische Subsystemverwaltung DSSM V3.6 und die Subsystemkatalog-Verwaltung SSCM V2.1 mit den dazugehörigen Kommandos und Anweisungen beschrieben.  
DSSM bietet die Möglichkeit, benutzereigene Subsystem-Konfigurationen tasklokal zu erstellen und zu verwalten.
- [29] **BS2000/OSD-BC V3.0**  
**Makroaufrufe an den Ablaufteil**  
Benutzerhandbuch
- Zielgruppe*  
Das Handbuch wendet sich an alle BS2000/OSD-Assembler-Programmierer.
- Inhalt*  
Das Handbuch enthält eine Zusammenstellung der Makroaufrufe an den Ablaufteil:
  - Binden und Laden
  - virtueller Speicher, Memory Pool, ESA
  - Task- und Programmsteuerung
  - ITC, Serialisierung, Eventing, DLM, Contingencys, STXIT
  - Meldungswesen, Accounting, JMS, TIAM, VTSU, ....Ausführliche Beschreibung aller Makroaufrufe in lexikalischer Reihenfolge mit Beispielen.  
Allgemeiner Lernteil über ITC, Serialisierung, Eventing, DLM, Contingencys, STXIT, virtueller Speicher, Memory Pool, ESA, ...
- [30] **SDF-P V2.0B (BS2000/OSD)**  
**Ergänzungsband**  
Beschreibung
- Zielgruppe*  
Der Ergänzungsband SDF-P V2.0B wendet sich an BS2000-Anwender und Systembetreuung.
- Inhalt*  
Im Ergänzungsband SDF-P V2.0B werden Erweiterungen und Korrekturen zum Handbuch bzw. zur Version SDF-P V2.0A beschrieben. Die wichtigsten Ergänzungen zu SDF-P V2.0A sind: Compiler-Unterstützung für S-Prozeduren, Steuerung von Eingabe-Kommandos aus einem Programm und Erweiterung beim Zugriff auf Variablen.

[31] **SPOOL V3.0A (BS2000/OSD)**

Benutzerhandbuch

*Zielgruppe*

Das Handbuch wendet sich an nichtprivilegierte Anwender, den RSO-Geräteverwalter und die Systembetreuung.

*Inhalt*

Es werden der Betrieb von SPOOL V3.0A mit den verfügbaren Kommandos, Makros, und System-Exits beschrieben. Die funktionellen Ergänzungen zu PRM V1.0A (Erstellung und Verwaltung von Druckressourcen) und zu Distributed Print Services V1.0A (Drucken im Rechnernetz) wurden berücksichtigt. Die ersten Kapitel informieren über BS2000-Schnelldrucker und Einstellung von Druckermerkmalen.

[32] **Spool & Print - Kommandos (BS2000/OSD)**

Benutzerhandbuch

*Zielgruppe*

Das Handbuch wendet sich an nichtprivilegierte Anwender, den Geräteverwalter, den Cluster-Vewalter, den SPOOL-Verwalter und die Systembetreuung.

*Inhalt*

Es werden die verfügbaren Kommandos von SPOOL V3.2A, Dprint V1.0D, RSO V3.0B und SPS V3.0A beschrieben. Kommandos zur Subsystemverwaltung und zur Auftragssteuerung sind nicht beschrieben.

[33] **SORT (BS2000/OSD)****SDF-Format***Zielgruppe*

- BS2000-Anwender
- Programmierer

*Inhalt*

Prinzipien, Funktionen und Anweisungen für das Sortieren und Mischen von Datensätzen (SDF-Format). Aufruf über Unterprogrammschnittstelle und Zugriffsmethode SORTZM. Ein Beispielkapitel führt den Anfänger in die Handhabung ein.

[34] **AID V2.2 (BS2000/OSD)****Testen unter POSIX**

Ergänzung

*Zielgruppe*

Das Handbuch richtet sich an C/C++-Programmierer.

*Inhalt*

- Beschreibung des Testens von C/C++-Programmen, die POSIX-Schnittstellen benutzen, bzw. die vollständig in der POSIX-Shell ablaufen.
- Anwendungsbeispiel
- Meldungen

- [35] **AID V2.3A (BS2000/OSD)**  
**Testen von C/C++-Programmen**  
Benutzerhandbuch

*Zielgruppe*

Das Handbuch richtet sich an C/C++-Programmierer.

*Inhalt*

Das Handbuch enthält die Beschreibung der AID-Kommandos und der C/C++-spezifischen Adreßoperanden zum symbolischen Testen von C/C++-Programmen. Es enthält Informationen zum Testen unter POSIX sowie zum Testen auf RISC-Anlagen und ausführliche Anwendungsbeispiele.

- [36] **WebServe (BS2000/OSD)**  
**WWW-Server auf BS2000/OSD**  
Benutzerhandbuch

*Zielgruppe*

Das Handbuch wendet sich an folgende Zielgruppen:

- Entscheider die planen, einen BS2000-Rechner als WWW-Server einzurichten
- BS2000-Anwender die einen BS2000-Rechner als WWW-Server mit dem Produkt *WebServe* einsetzen.

*Inhalt*

Das Handbuch beschreibt das fachliche Umfeld und den Einsatz eines BS2000-Rechners als WWW-Server mit dem Produkt *WebServe*. Nach einer kurzen Einführung in das Internet und das World Wide Web werden einige Anwendungsszenarien für einen WWW-Server im BS2000 beschrieben. Der umfangreichste Teil befasst sich mit der Installation und Konfiguration von *WebServe*. Der Anhang enthält unter anderem eine HTML-Kurzreferenz.

- [37] **WebTransactions**  
**Geschäftsrelevante Anwendungen im Internet und Intranet**  
Kurzinformation

- [38] **WebTransactions**  
**Transaktionsorientierter Web-Zugriff für openUTM-Anwendungen**  
Benutzerhandbuch

- [39] **WebTransactions für OSD und MVS**  
**Web-Anschluß für Dialoganwendungen unter OSD und MVS**  
Benutzerhandbuch

(Die Handbücher zu *WebTransactions* finden Sie auf dem BS2000-WWW-Server unter der URL [http://bs2www.mch.sni.de/webtrans/manual\\_d.htm](http://bs2www.mch.sni.de/webtrans/manual_d.htm).

Sie können als Postscriptdatei oder HTML-Dokument heruntergeladen oder direkt angesehen werden. Im Druckschriftenverzeichnis sind die Handbücher nicht enthalten.)

- [40] **TCP-IP-AP (TRANSDATA, BS2000)**

Benutzerhandbuch

*Zielgruppe*

BS2000-Systemverwalter und FTP- bzw. TELNET-Benutzer

*Inhalt*

Beschreibung der TCP/IP-Anwendungen. Anleitung für das Arbeiten mit FTP und TELNET. Kommandos für FTP, TELNET und die Administrationsschnittstelle.

- [41] **TCP-IP-SV V1.0 (BS2000/OSD)**

Benutzerhandbuch

*Zielgruppe*

Das Handbuch richtet sich an Netzplaner, -generierer und -verwalter, die in BS2000-Systemen BCAM mit TCP/IP-Protokollen betreiben.

*Inhalt*

Das Handbuch stellt die Resolver-Funktionalität von DNS und die Server-Funktionalität der Komponenten BOOTP/DHCP und TFTP des Produkts TCP-IP-SV vor. Es werden Installation, Administration und Betrieb der einzelnen Komponenten beschrieben. Die Logging- und Diagnose-Möglichkeiten sind Thema eines eigenen Kapitels.

- [42] **TransView SNMP Basic Agent BS2000 V2.0/V2.1**

**TransView SNMP Standard Collection BS2000 V2.0/V2.1**

Benutzerhandbuch

*Zielgruppe*

Das Handbuch wendet sich an Netzverwalter, -operateure und Systemverwalter, die BS2000-Systeme in ein SNMP-basiertes Management integrieren bzw. ein solches System bedienen wollen.

*Inhalt*

Dieses Handbuch beschreibt einerseits die Einbettung von TV-SBA-BS2, TV-SSC-BS2, TV-SSA-SM2 und TV-SSA-UTM in BS2000/OSD und die zum Betrieb notwendigen Installations- und Konfigurationsschritte sowie den Betrieb selbst. Die zur Überwachung notwendigen Agenten und ihre MIBs werden detailliert vorgestellt. Andererseits wird die Installation und Konfiguration der entsprechenden Management-Anwendungen auf der TransView Management-Station dargestellt.



---

# Stichwörter

\$HOME/.profile 50  
\$HOME/.rhosts 44  
.profile-Datei 56  
/etc/group 121  
/etc/inetd.conf 49  
/etc/passwd 117  
/etc/profile 43, 50  
/etc/rc0.d 109  
/etc/rc2.d 109  
/etc/vfstab 111  
/var/adm/messages 84  
/var/sadm/pkg/instlog 84

## A

Abrechnungsnummer für rlogin verwalten 125  
ADD-USER 127  
AID 23  
Aliasname 188  
alphanum-name (Datentyp) 189  
Anweisungszeile 95, 96, 97, 99, 100  
Anwendung portieren 16  
append 91, 97  
ASCII 16, 29  
ASCII-EBCDIC-Konvertierung 27, 90  
Aufrufargument 52  
Aufrufertask 38  
Ausführberechtigung 42  
aushängen (Dateisystem) 111  
Authentisierung 39

## B

Backup-Server 18  
Basis-Shell 179  
BCAM 69

Behälterdatei 17, 26  
    Zugriffsschutz 41  
Beispielsitzung (POSIX-Shell) 56  
Benutzerattribute, siehe POSIX-Benutzerattribute  
Benutzerdaten 40  
Benutzerdatenverwaltung 39  
Benutzerkatalog  
    Eintrag erstellen 127  
    Informationen per Programm lesen 126  
    Informationen über Einträge ausgeben 156  
Benutzerkennung 41  
    Benutzernummer zuordnen 119  
Benutzernummer 117, 119  
    0 (Null) 117  
    einer Benutzerkennung zuordnen 119  
Benutzerverwaltung, siehe POSIX-Benutzerverwaltung  
Bereitzeichen 50, 56  
Bibliotheksfunktionen 1  
BINDANY 106  
Bindelademodul 33  
Binder-Lader-System 59  
Blockterminal 21, 46, 49  
BLS 23  
BS2000 als Server 18  
BS2000/OSD-Versionswechsel 86, 87  
BS2000-Datei 27  
BS2000-Gruppen 40  
BS2000-Programmschnittstellen 1, 53  
BS2000-Softwarereprodukte 23  
BUFHWM 103

**C**  
C++-Compiler 23, 60  
Cache-Puffer 103  
cat (Zusatz zu Datentypen) 200  
cat-id (Datentyp) 189  
C-Bibliothek 1, 21  
C-Bibliotheksfunktionen 2  
C-Compiler 23, 60  
chmod 43  
chown 123  
C-Laufzeitsystem 45  
Client-Server-Architekturen 13  
command-rest (Datentyp) 189

compl (Zusatz zu Datentypen) 195  
composed-name (Datentyp) 189  
copy 25  
copy-on-access-Mechanismus 38  
corr (Zusatz zu Datentypen) 200, 201  
cp 25  
CRTE 1, 2, 23, 61, 77, 79  
c-string (Datentyp) 189  
CTRL-Taste 50

## D

Dämon 38  
date (Datentyp) 189  
Datei  
    BS2000 27  
    POSIX 24, 27  
Dateien kopieren 27  
Dateien, ferne 30  
Dateisystem  
    auf Konsistenz prüfen 112  
    aushängen 111  
    einhängen 111  
    hierarchisches 17, 24, 25  
    POSIX 17, 24  
    UNIX 17, 24  
    verwalten 111

Dateisystemparameter 103

Dateiverzeichnis 24  
    angelegt 182  
    root 24

Datenhaltung, verteilte 19

Datenserver 18

Datentypen SDF 186, 189  
    Zusätze 186

DBLPOOL 51, 106

DBLSTATE 51, 106

DCE 23

DCE (BS2000) 75

debug 51

delete 92, 97

device (Datentyp) 189

Dokumentation zu POSIX 2

DSSM 33

### E

EBCDIC 16, 27, 29  
EDT 2, 23, 53, 62  
Einführung in POSIX 9  
Einhängepunkt 111  
Eintrag im Benutzerkatalog erstellen 127  
Environment-Variable siehe Umgebungsvariable  
Erstinstallation 85, 88  
etc/dfstab 112  
etc/vfstab 112  
exec **36**, 38  
execute 42

### F

FDFLUSHR 103  
ferne Dateien 30  
ferne Rechner 30, 67  
filename (Datentyp) 190  
File-Server 18  
fixed (Datentyp) 189  
FLCKREC 103  
fork **35**, 36, 38  
fsck 112  
fsflush 103  
FT-BS2000 23, **64**  
FTOS-BS2000 64  
full-filename  
    siehe Datentyp file-name 190

### G

gemischtes Programm 53  
gen (Zusatz zu Datentypen) 200  
Gerätedateien, angelegte 183  
Gruppennummer **117**, 121  
Gruppenverwalter 114  
Gruppenverwaltung **40**

### H

Handbuch-Konzept 4  
Hardware-Voraussetzungen für POSIX 21  
Hash-Anker 103  
heterogenes Netzwerk 1

hierarchisches Dateisystem 17, 24, 25  
Holdertask 38  
HSMS 23, 53, **65**

## I

iconv 29  
Identifikationszeichen 43  
Identifikationszeile **95**, 96, 97, 99, 100  
IEEE 1, 11  
in@telnet (telnet-Dämon) starten 49  
Index  
    global 198  
    Konstruktionszeichenfolge 198  
    platzhalter-spezifisch 198  
    -Schreibweise 199  
inetd-Dämon 109  
Informationsdatei, siehe POSIX-Informationsdatei  
Informationstechnologie (Anforderungen) 10  
init 34  
Init-Prozeß 34  
Installationsprogramm 80  
Installationsschritte 78  
Installieren  
    des POSIX-Subsystems 89  
integer (Datentyp) 191  
Interoperabilität 17  
IO\_CONVERSION (Umgebungsvariable) 27, 29

## K

Katalogeintrag  
    eines Benutzers ändern 139  
Kommando-Returncodes 202  
Kommandos in der POSIX-Shell eingeben 52  
Kommentar (Benutzerattribut) 118  
Kommentarzeile 95  
Konstruktionsangabe 199  
Konstruktionszeichenfolge 198  
Konvertierung (ASCII-EBCDIC) 27, 90  
Konzept (dieses Handbuchs) 4  
Kopieren von Dateien 27  
Kurzname 188

### L

Lader von POSIX 51  
Leseberechtigung 42  
Lieferumfang von POSIX 77  
Lieferverfahren für POSIX-Produkte 81  
link 25  
LINKER 51  
ln 25  
LOADER siehe POSIX-Lader  
Logging-Datei 84  
Login-Verzeichnis (Benutzerattribut) 118  
lokaler Rechner 30  
low (Zusatz zu Datentypen) 195

### M

Maincode 202  
man (Zusatz zu Datentypen) 200, 201  
mandatory (Zusatz zu Datentypen) 201  
MAXTIMERC 106, 109  
Metasprache **8**  
Metasyntax SDF 186, 187  
Mini-POSIX-Dateisystem 31  
mkdir 123  
modify 92, 97  
MODIFY-LOGON-PROTECTION 129  
MODIFY-POSIX-USER-ATTRIBUTES 131  
MODIFY-POSIX-USER-DEFAULTS 136  
MODIFY-USER-ATTRIBUTES 139  
mount 30, 67, 111, 112  
mountall 112  
mounten, siehe einhängen  
move 25  
MSGMAP 104  
MSGMAX 104  
MSGMNB 104  
MSGMNI 104  
MSGSEG 105  
MSGSSZ 104  
MSGTQL 104  
mv 25

**N**

Nachrichten-Warteschlange 104  
name (Datentyp) 191  
NAUTOUP 103  
NBUF 103  
Netzwerk, heterogenes 1  
NFS 17, 23, 30, 67, 67  
NHBUF 103  
NOPTY 106  
NOSTTY 106  
NOTTY 106  
NPROC 103  
NRNODE 104

**O**

odd (Zusatz zu Datentypen) 200  
offene Systeme 11  
offenes BS2000 11  
Offenheit 11, 13  
openFT (BS2000) 64  
OSF 11

**P**

PAMCONV 31  
Parametertabelle 33  
partial-filename (Datentyp) 192  
path-compl (Zusatz zu Datentypen) 195  
Performance 33  
Pfadname 27  
PID 35  
Pipe 37  
Portabilität 11, 16  
Portieren einer Anwendung 16  
PORTMON 106  
POSIX  
    beenden 110  
    Definition 1  
    Lieferumfang 77  
    Softwareprodukt 1  
    starten 108  
    Status anzeigen 144  
    steuern 107  
POSIX-ADDON-LIB 77  
POSIX-ADMINISTRATION 114

POSIX-Anwendung 13  
POSIX-BC 77  
POSIX-Benutzer  
  löschen 125  
POSIX-Benutzerattribute 40, 117  
  ändern 131  
  anzeigen 145  
  Standardwerte ändern 136  
  Standardwerte anzeigen 153  
POSIX-Benutzerverwaltung 113  
POSIX-Datei 17, 24, 27  
  mit EDT bearbeiten 62  
POSIX-Dateisystem 13, **17**, 24  
  aushängen 112  
  einhängen 111, 112  
  verwalten 91, 97, 107  
POSIX-Dokumentation (Konzept) 2  
POSIX-Einführung 9  
posix-filename (Datentyp) 192  
POSIX-Gruppen 40, 117  
POSIX-Gruppenkatalog 121  
POSIX-HEADER 77  
POSIX-Informationsdatei 33, **101**  
POSIX-Installationsprogramm 26, 41, 80  
  im Dialog 88  
  im Stapelbetrieb 95  
POSIX-Lader 51  
POSIX-Laders 106  
POSIX-NSL (Release Unit) 70  
posix-pathname (Datentyp) 192  
POSIX-Platzhalter 196  
POSIX-Programmschnittstellen 1, 30, **53**  
POSIX-Prozeßverwaltung 34  
POSIX-SH 77  
POSIX-Shell 1, 36, **45**  
  Kommandoeingabe 52  
  starten 157  
  Zugang 47  
POSIX-SOCKETS 69, 77  
POSIX-Standard 1  
  Vorteile 16

POSIX-Subsystem 1, 32  
einrichten 96  
installieren 89  
verwalten 33  
POSIX-Versionswechsel 86  
POSIX-Verwalter 114  
POSPRTS 77  
Print-Server 18  
Privileg 114  
Gruppenverwalter 114  
POSIX-ADMINISTRATION 114  
SECURITY-ADMINISTRATION 114  
USER-ADMINISTRATION 114  
product-version (Datentyp) 193  
Programm (Benutzerattribut) 118  
Programm, gemischtes 53  
Programmpakete  
entfernen 88, 94, 100  
hinzufügen 88, 92, 99  
Programmschnittstellen 53  
BS2000 1  
POSIX 1  
Prompt, siehe Bereitzeichen  
protokollieren, POSIX-Installation 84  
Prozeßidentifikation 35  
Prozeßumgebung 35, 36, 38  
Prozeßverwaltung, siehe POSIX-Prozeßverwaltung  
P-Tasten belegen 50

## **Q**

quotes (Zusatz zu Datentypen) 201

## **R**

rc-Beendigungsprozeduren 106  
rcp 44  
rc-Prozeduren 109  
read 42  
Rechner  
    ferner 30, 67  
    lokaler 30  
Rechnerverbund 15  
Resourcemap 104, 105  
rhosts-Datei 44

rlogin 44, 48, 109  
    Systemzugang verwalten 124  
root 27  
Root-Berechtigung 117  
root-Verzeichnis 24  
rsh 44

### S

Schnittstellenplattform SIA 11  
Schreibberechtigung 42  
Schutzattribute  
    ändern 129  
    anzeigen 143  
    vereinbaren 141  
Schutzbit-Maske 43  
Schutzbits 42  
SCI 82  
SDF-A 23, 67  
SECOS 23, 48, 113, 124  
SECURITY-ADMINISTRATION 114  
SEGMAPSZ 104  
SEMAEM 105  
Semaphor 104, 105  
SEMMAP 105  
SEMMNI 105  
SEMMNS 105  
SEMMNU 105  
SEMMSL 105  
SEMOPM 105  
SEMUME 105  
SEMVMX 105  
sep (Zusatz zu Datentypen) 200  
Server 18  
SET-LOGON-PROTECTION 141  
share 30, 67, 112  
shareall 112  
Shell, siehe POSIX-Shell  
Shell-Prozedur 46  
Shell-Variable siehe Umgebungsvariable  
Shell-Variablen 50  
SHMMAX 105  
SHMMIN 105  
SHMMNI 105  
SHMSEG 105

SHOW-JOB-STATUS 51  
SHOW-LOGON-PROTECTION 143  
SHOW-POSIX-STATUS 144  
SHOW-POSIX-USER-ATTRIBUTES 145  
SHOW-POSIX-USER-DEFAULTS 153  
SHOW-USER-ATTRIBUTES 156  
SIA-Schnittstellenplattform 11  
Sicherheitskonzept 39  
SINIX-Systemkern 32  
SOCKETS 23, 106  
SOCKETS/XTI **69**  
Software Configuration Inventory (SCI) 82  
Softwareprodukte 23  
Sohnprozeß 35  
SORT 23  
Speicherbereich 105  
SPOOL 23, **70**  
SRMUINF 126  
SRPM 39, 44, 48, 113  
Standard  
    POSIX 1  
    UNIX95 1  
    XPG4 1  
    XPG4.2 1, 61  
Standard-Benutzernummer 119  
Standardschnittstellen 12  
Standardwerte  
    der Steuerparameter ändern 103  
    für POSIX-Benutzerattribute ändern 136  
    für POSIX-Benutzerattribute anzeigen 153  
    für POSIX-Benutzerattribute verwalten 124  
START-POSIX-SHELL 157  
Steuerparameter 33, 101  
    für die Interprozeßkommunikation 104  
    Standardwerte ändern 103  
STOP-SUBSYSTEM POSIX 110  
structured-name (Datentyp) 193  
Subcode1 202  
Subcode2 202  
Subsystem POSIX 1, 32  
SYSROOT **114**, 117, 124  
SYSSRPM 47, 117

Systemkern SINIX 32

Systemparameter, allgemeine 103

Systemplattform 12

Systemprozeß 38

### T

TCP-IP-AP 23, 74

TCP-IP-SV 23, 75

telnet 49

telnet-Dämon starten 49

temp-file (Zusatz zu Datentypen) 200

Terminal (Unterstützung) 21

Terminal-Emulation 47

text (Datentyp) 193

time (Datentyp) 193

TLI 23

TLI-Schnittstellen 70

Transaktionsverarbeitung, verteilte 12

TransView SNMP-Basic-Agent BS2000 23, 76

Tuningparameter, siehe Steuerparameter

### U

UFSNINODE 104

umask 43

Umgebungsvariable

    IO\_CONVERSION 27, 29

umount 111, 112

umountall 112

under (Zusatz zu Datentypen) 195

Undo-Operation 105

UNIX95-Standard 1

UNIX-Dateisystem 17, 24

unshare 112

unshareall 112

Upgrade-Installation 86

user (Zusatz zu Datentypen) 201

USER-ADMINISTRATION 114

### V

var-Dateisystem 111

Vaterprozeß 35

Vater-Sohn-Beziehung 34, 36

Verarbeitung, verteilte 13, 20

Vernetzung heterogener Systeme 10

vers (Zusatz zu Datentypen) 201  
Versionswechsel  
    BS2000/OSD 86, 87  
    POSIX 86  
verteilte Datenhaltung 19  
verteilte Transaktionsverarbeitung 12  
verteilte Verarbeitung 13, 20  
Verwaltung des POSIX-Subsystems 33  
Verwaltungsdateien, angelegte 184  
Voraussetzungen für Installation 78  
Voreinstellungen (Benutzerumgebung) 50  
Vorteile des POSIX-Standards 16  
vsn (Datentyp) 193

## W

WebServe installieren 85  
wild(n) (Zusatz zu Datentypen) 196  
with (Zusatz zu Datentypen) 195  
without (Zusatz zu Datentypen) 200  
write 42

## X

X/OPEN 1, 11  
XPG4.2-Standard 1, 61  
XPG4-Standard 1  
x-string (Datentyp) 194  
x-text (Datentyp) 194

## Z

Zeichenterminal 21, 46, 49  
Zielgruppen (dieses Handbuchs) 4  
Zugang zur POSIX-Shell 47  
Zugangsberechtigung verwalten 124  
Zugriffsberechtigung 42  
Zugriffsschutz 41  
Zugriffsschutz (Behälterdatei) 41  
Zusätze zu Datentypen 186, 195



# Inhalt

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Konzept der POSIX-Dokumentation	2
1.2	Zielgruppen des Handbuchs	4
1.3	Konzept des Handbuchs	4
1.4	Änderungen gegenüber der vorigen Ausgabe	6
1.5	Verwendete Metasprache	8
<b>2</b>	<b>Einführung in POSIX</b>	<b>9</b>
2.1	POSIX im BS2000/OSD	9
	Neue Anforderungen an die Informationstechnologie	10
	Eine Welt offener Systeme	11
	Offenes BS2000 - BS2000/OSD	11
	Offenheit durch Client-Server-Architekturen	13
	BS2000/OSD bringt die UNIX- und die BS2000-Welt zusammen	15
	Vorteile des POSIX-Standards	16
	Bestandteile von POSIX	21
	Hardware-Voraussetzungen für POSIX	21
	Unterstützung von Terminals	21
	An POSIX angepaßte BS2000-Softwareprodukte	23
2.2	POSIX-Dateisystem	24
	Vorteile eines hierarchischen Dateisystems	25
	Ablage von POSIX-Dateisystemen in Behälterdateien	26
	Vorteile durch das Anlegen mehrerer POSIX-Dateisysteme	26
	Konventionen für Namen von POSIX-Dateien und Dateiverzeichnissen	27
	Kopieren und Konvertieren von Dateien	27
	Zugriff auf POSIX-Dateisysteme im BS2000	29
	Zugriff auf POSIX-Dateien	30
	Zugriff auf ferne Dateien	30
	Migration von Mini-POSIX-Dateisystemen	31
2.3	POSIX als Subsystem im BS2000	32
	Verwaltung des Subsystems POSIX durch DSSM	33
	POSIX-Prozeßverwaltung	34

2.4	Sicherheitskonzept .....	39
	Benutzerdatenverwaltung .....	39
	Gruppenverwaltung .....	40
	Zugriffsschutz für Behälterdateien .....	41
	Zugriffsschutz für Dateien und Dateiverzeichnisse .....	41
	Zugangsschutz bei Zugang über einen fernen Rechner .....	44
<b>3</b>	<b>Arbeiten mit POSIX .....</b>	<b>45</b>
3.1	POSIX-Shell .....	45
3.1.1	Zugang zur POSIX-Shell .....	47
3.1.2	Besonderheiten für das Arbeiten mit der POSIX-Shell .....	50
3.1.3	POSIX-Lader .....	51
3.1.4	Kommandos von der POSIX-Shell aus eingeben .....	52
3.2	POSIX-Programmschnittstellen .....	53
3.2.1	Einschränkungen für Programme mit gemischter Funktionalität .....	54
3.2.2	Einschränkungen für Makroaufrufe .....	55
3.2.3	Vererbung .....	55
3.3	Beispielsitzung .....	56
<b>4</b>	<b>BS2000-Softwareprodukte im Umfeld von POSIX .....</b>	<b>59</b>
4.1	Binder-Lader-System ab BS2000/OSD-BC V2.0 .....	59
4.2	C/C++-Compiler ab V2.2 .....	60
4.3	EDT ab V16.5 .....	62
4.4	File-Transfer-Produkte (FT-BS2000, FTOS-BS2000, <i>openFT</i> ) .....	64
4.5	HSMS / HSMS-SV ab V2.0B .....	65
4.6	NFS ab V1.2 .....	67
4.7	SDF-A ab V4.0 .....	67
4.8	SECOS ab V2.1 .....	68
4.9	SOCKETS/XTI (POSIX-SOCKETS) ab V1.1 .....	69
4.10	SPOOL ab V3.0 .....	70
4.11	TLI (POSIX-NSL) ab V1.0 .....	70
4.12	AID ab V2.2 .....	71
4.13	SORT ab V7.5 .....	72
4.14	TCP-IP-AP ab V2.0 .....	74
4.15	TCP-IP-SV .....	75
4.16	DCE (BS2000) .....	75
4.17	TransView SNMP-Basic-Agent BS2000 .....	76
<b>5</b>	<b>POSIX installieren .....</b>	<b>77</b>
5.1	Lieferumfang .....	77
5.2	Installationsschritte .....	78
5.3	POSIX-Installationsprogramm .....	80
5.3.1	Liefer- und Installationsverfahren für POSIX-Programmpakete .....	81
5.3.2	Erstinstallation .....	85

5.3.3	Upgrade-Installation . . . . .	86
5.3.4	POSIX-Installationsprogramm im Dialog . . . . .	88
	Install POSIX subsystem (Subsystem POSIX neu einrichten) . . . . .	89
	Administrat. POSIX filesystems (POSIX-Dateisysteme verwalten) . . . . .	91
	Install packages on POSIX (POSIX-Programmpakete hinzufügen) . . . . .	92
	Delete packages from POSIX (POSIX-Programmpakete entfernen) . . . . .	94
5.3.5	Automatisierter Ablauf des POSIX-Installationsprogramms . . . . .	95
	Install POSIX subsystem (Subsystem POSIX neu einrichten) . . . . .	96
	Administrat. POSIX filesystems (POSIX-Dateisysteme verwalten) . . . . .	97
	Install Packages on POSIX (Programmpakete hinzufügen) . . . . .	99
	Delete Packages from POSIX (Programmpakete entfernen) . . . . .	100
5.4	POSIX-Informationsdatei . . . . .	101
<b>6</b>	<b>POSIX steuern, Dateisysteme verwalten</b> . . . . .	<b>107</b>
6.1	POSIX starten . . . . .	108
6.2	POSIX beenden . . . . .	110
6.3	Dateisysteme verwalten . . . . .	111
6.3.1	Lokale POSIX-Dateisysteme verwalten . . . . .	112
6.3.2	Verteilte Dateisysteme verwalten . . . . .	112
6.3.3	Dateisystem auf Konsistenz prüfen . . . . .	112
<b>7</b>	<b>POSIX-Benutzer verwalten</b> . . . . .	<b>113</b>
7.1	Übersicht über Privilegien und Aufgaben . . . . .	114
7.2	POSIX-Benutzerattribute vergeben . . . . .	117
7.3	Einer BS2000-Benutzerkennung eine individuelle Benutzernummer zuordnen . . . . .	119
7.4	BS2000- und POSIX-Gruppen verwalten . . . . .	120
7.5	Neue POSIX-Benutzer eintragen . . . . .	122
7.6	Standardwerte für POSIX-Benutzerattribute festlegen . . . . .	124
7.7	Zugangsberechtigung für Benutzer eines fernen Rechners erteilen . . . . .	124
7.8	Abrechnungsnummer für Systemzugang über fernen Rechner eintragen . . . . .	125
7.9	POSIX-Benutzer löschen . . . . .	125
7.10	Benutzerinformationen per Programm lesen . . . . .	126
<b>8</b>	<b>BS2000-Kommandos für POSIX</b> . . . . .	<b>127</b>
	ADD-USER	
	Benutzereintrag im Benutzerkatalog erstellen . . . . .	127
	MODIFY-LOGON-PROTECTION	
	Schutzattribute ändern . . . . .	129
	MODIFY-POSIX-USER-ATTRIBUTES	
	POSIX-Benutzerattribute ändern . . . . .	131
	MODIFY-POSIX-USER-DEFAULTS	
	Standardwerte für POSIX-Benutzerattribute ändern . . . . .	136
	MODIFY-USER-ATTRIBUTES	
	Katalogeintrag eines Benutzers ändern . . . . .	139

SET-LOGON-PROTECTION	141
Schutzattribute vereinbaren	141
SHOW-LOGON-PROTECTION	143
Schutzattribute anzeigen	143
SHOW-POSIX-STATUS	144
POSIX-Status anzeigen	144
SHOW-POSIX-USER-ATTRIBUTES	145
POSIX-Benutzerattribute anzeigen	145
SHOW-POSIX-USER-DEFAULTS	153
Standardwerte für POSIX-Benutzerattribute anzeigen	153
SHOW-USER-ATTRIBUTES	156
Informationen über die Einträge im Benutzerkatalog ausgeben	156
START-POSIX-SHELL	157
POSIX-Shell zur Verfügung stellen	157
<b>9      POSIX-Meldungen</b>	<b>159</b>
<b>10     Anhang</b>	<b>175</b>
10.1    Privilegien bei POSIX	176
10.2    Kommandoumfang der Basis-Shell	179
10.3    Dateiverzeichnisse, die bei einer Erstinstallation angelegt werden	182
10.4    Gerätedateien, die bei einer Erstinstallation angelegt werden	183
10.5    Verwaltungsdateien, die bei einer Erstinstallation angelegt werden	184
10.6    Tuning-Maßnahmen	185
10.7    SDF-Syntaxbeschreibung	186
10.8    Kommando-Returncodes	202
<b>Abkürzungen</b>	<b>203</b>
<b>Fachwörter</b>	<b>207</b>
<b>Literatur</b>	<b>225</b>
<b>Stichwörter</b>	<b>239</b>

---

# POSIX (BS2000/OSD)

## Grundlagen für Anwender und Systemverwalter

### *Zielgruppe*

BS2000-Systemverwalter, POSIX-Verwalter, BS2000-Benutzer,  
Benutzer von UNIX-/SINIX-Workstations

### *Inhalt*

- Einführung und Arbeiten mit POSIX
- BS2000-Softwareprodukte im Umfeld von POSIX
- POSIX installieren
- POSIX steuern und Dateisysteme verwalten
- POSIX-Benutzer verwalten
- BS2000-Kommandos für POSIX

**Ausgabe:** **April 1998**

gültig für POSIX-BC V1.1, V3.0 und V3.1

**Datei:** **posix\_g.pdf**

Copyright © Siemens Nixdorf Informationssysteme AG, 1998.

Alle Rechte vorbehalten.

Liefermöglichkeiten und technische Änderungen vorbehalten.

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