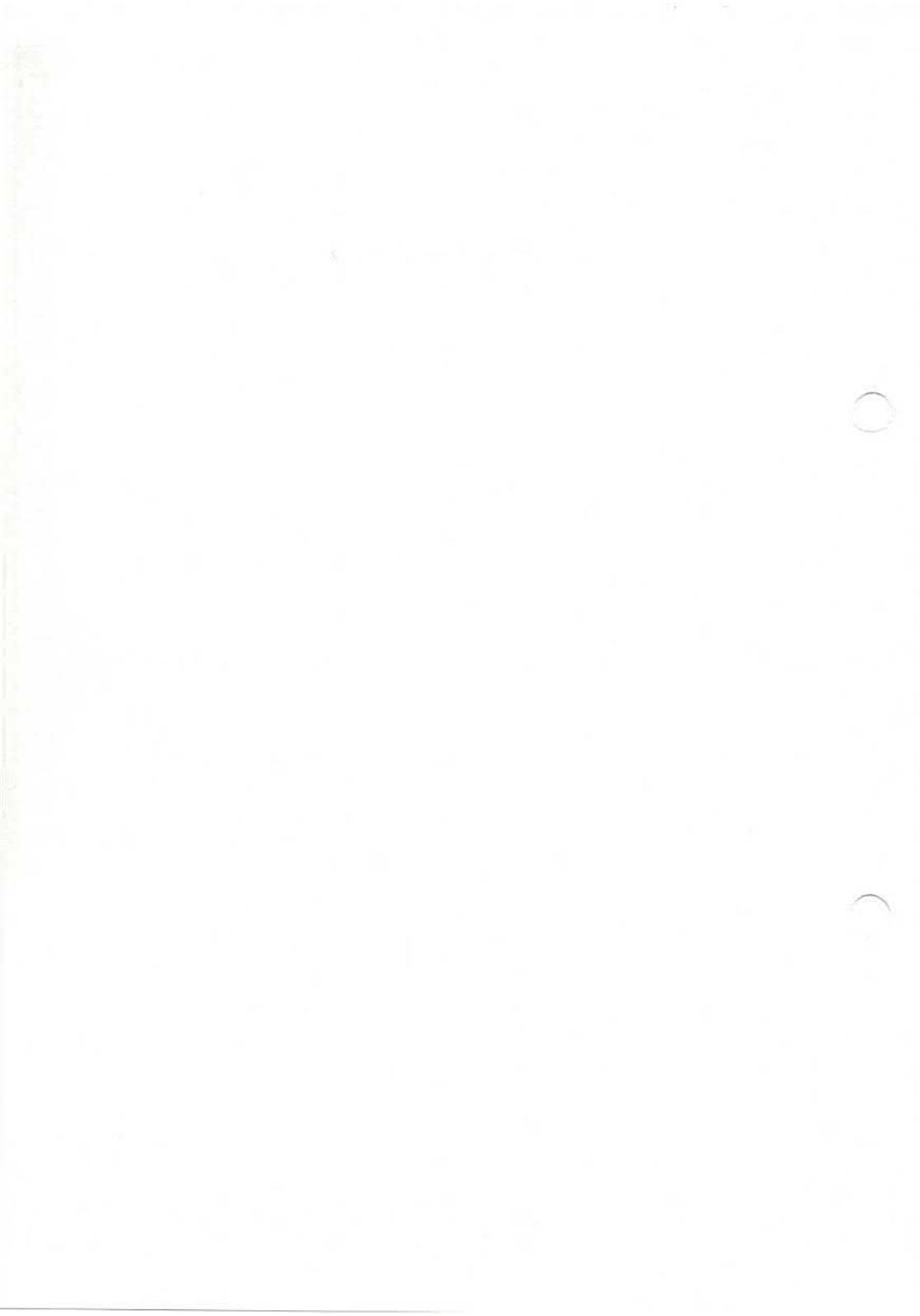


TARGON[®] /31

Systemadministration



Einleitung

Kurzbeschreibung Hardware TARGON /31

Booten der TARGON /31

Systemverwaltung

Das Spool-System

Anhang 1: Administrator-Kommandos

Anhang 2: Gerätedateien

Anhang 3: Dateiformate

Anhang 4: Stichwortverzeichnis

1

2

3

4

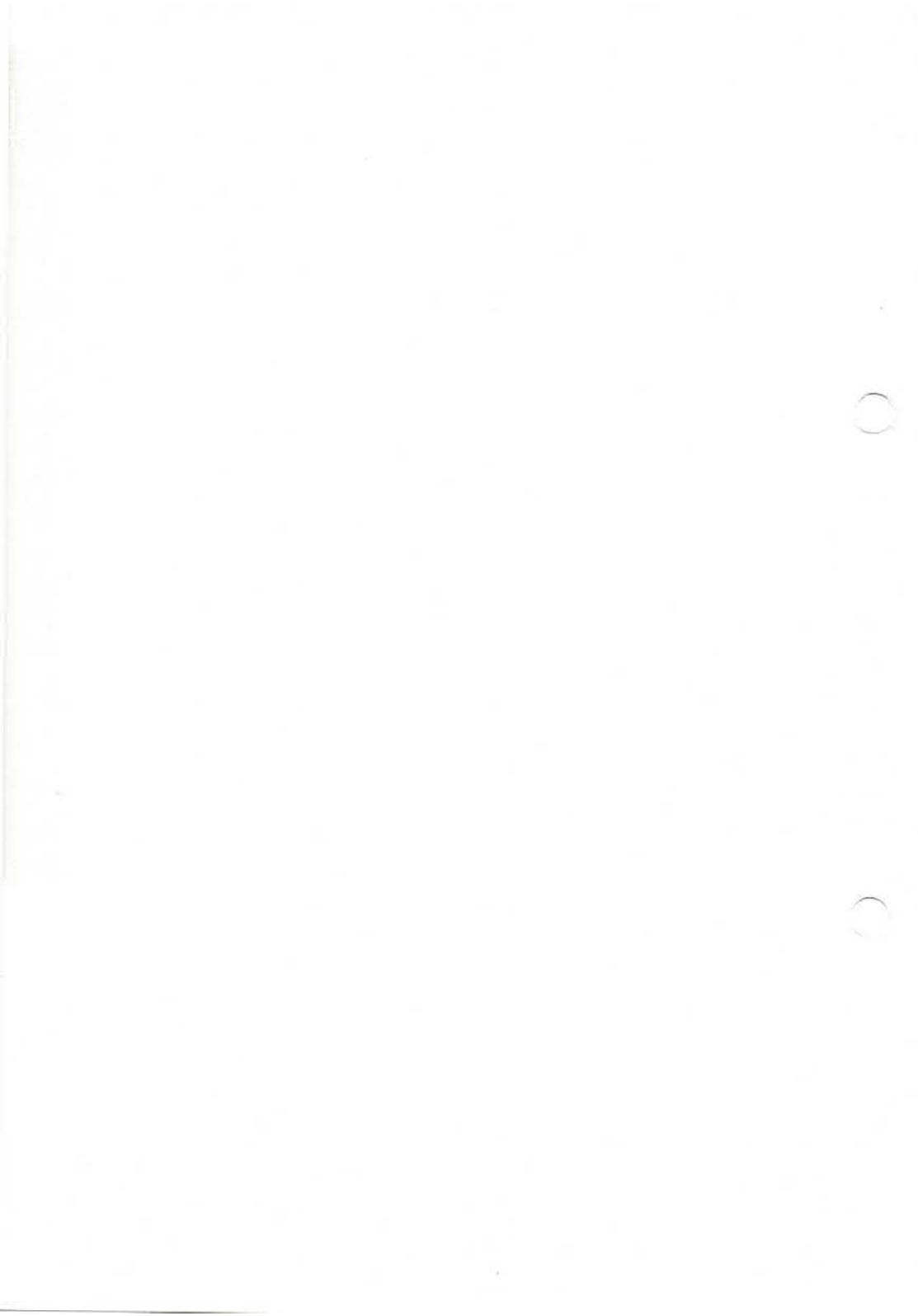
5

A1

A2

A3

A4



Organisationsblatt

Organisationsblatt

Dieses Blatt gibt eine Übersicht über alle Änderungen, die seit der ersten Auflage an diesem Handbuch durchgeführt wurden. Es wird bei jeder Änderungsmitteilung mitgeliefert und ist jeweils auszutauschen.

Erstauflage:

1.11.86

Rel. 2

Änderungswünsche/Fehler

Änderungswünsche/Fehler

Sollten Ihnen bei der Benutzung dieses Teils der Systemliteratur Fehler aufgefallen sein oder sollten Sie Vorschläge zur Verbesserung des Handbuchs haben, so bitten wir Sie, diese schriftlich zu formulieren und an folgende Anschrift zu schicken:

Nixdorf Computer AG
Abt. ZSI
Fürstenallee 7

D-4790 Paderborn

Inhaltsverzeichnis

1	Einleitung	1-1
2	Kurzbeschreibung Hardware TARGON /31	2-1
2.1	TARGON /31, Modell 30.....	2-1
2.1.1	CPU-Ausstattung, Modell 30.....	2-2
2.1.2	Prozessorausstattung, Modell 30 und Modell 50.....	2-3
2.1.3	Speicherausstattung, Modell 30 und Modell 50.....	2-4
2.1.4	System-Peripherie-Prozessor (SPP).....	2-5
2.1.5	Input/Output-Prozessor (IOP), Modell 30 und Modell 50.....	2-6
2.1.6	Input/Output-Controllerausstattung, Modell 30 und Modell 50.....	2-7
2.1.6.1	Terminal-Communication-Controller (TCC).....	2-8
2.1.6.2	Der Multifunktions-Controller (MFC).....	2-10
2.1.7	Übersicht Modell 30.....	2-11
2.1.7.1	Grundausrüstung.....	2-11
2.1.7.2	Maximalausstattung.....	2-12
2.2	TARGON /31, Modell 50.....	2-13
2.2.1	CPU-Ausstattung, Modell 50.....	2-14
2.2.2	System-Peripherie-Prozessor (SPP).....	2-15
2.2.3	Disk/Tape-Prozessor, Modell 50 (DTP).....	2-16
2.2.4	Übersicht Modell 50.....	2-17
2.2.4.1	Grundausrüstung.....	2-17
2.2.4.2	Maximalausstattung.....	2-18
2.3	Peripherie.....	2-19
3	Booten der TARGON /31	3-1
3.1	Boot-Phasen.....	3-1
3.2	Startsmall.....	3-1
3.3	First Level Boot.....	3-3
3.3.1	Autoboot.....	3-4
3.3.2	Standalone.....	3-5
3.3.3	Manual Boot.....	3-6
3.3.4	Dump Image.....	3-6
3.4	Second Level Boot.....	3-7
3.4.1	Booten des Betriebssystems.....	3-8
3.4.2	Sichern des Root-Dateisystems.....	3-9
3.4.3	Einlesen des Root-Dateisystems.....	3-10
3.4.3.1	Inhalt des Root-Dateisystems.....	3-11
3.4.3.2	Inhalt des Dateisystems /usr.....	3-12
3.4.4	Auswahl des Boot-Laufwerks.....	3-13

Inhaltsverzeichnis

3.4.5	Sichern des Second Level Boot.....	3-13
3.4.6	Platten formatieren.....	3-13
4	Systemverwaltung	4-1
4.1	Verwaltungsdateien.....	4-1
4.2	Eintragen neuer Benutzer.....	4-2
4.3	Datenschutzmechanismen.....	4-3
4.4	Sicherung und Wiederherstellung von Daten.....	4-5
4.4.1	Schutzmechanismen für Benutzerdateien.....	4-5
4.4.2	Programme zur Sicherung von Dateisystemen.....	4-6
4.5	Hilfreiche Programme für den Systemverwalter.....	4-7
4.5.1	Kommunikation mit den Benutzern.....	4-7
4.5.2	Überwachung der Plattenverwendung.....	4-7
4.5.3	Verkleinern von Verzeichnissen.....	4-9
4.5.4	Reinigen der administrativen Log-Dateien.....	4-10
4.6	Zeitplan des Systemverwalters.....	4-11
5	Das Spool-System	5-1
5.1	Konfiguration.....	5-1
5.2	Kommandos.....	5-2
5.2.1	Benutzerkommandos.....	5-2
5.2.2	Administrator-Kommandos.....	5-3
5.3	Die Installation des Spool-Systems.....	5-4
5.3.1	Vorsichtsmaßnahmen.....	5-4
5.4	Die Konfiguration des Spoolers mit lpadmin.....	5-5
5.4.1	Druckerkonfiguration.....	5-5
5.4.1.1	Nicht-optionale Parameter.....	5-5
5.4.1.2	Optionale Parameter.....	5-6
5.4.1.3	Beispiele zur Druckerkonfigurierung.....	5-7
5.4.2	Änderung von Druckern oder Druckerklassen.....	5-8
5.4.3	Beispiele zur Änderung von Druckern.....	5-9
5.4.4	Einrichten des Standard-Systemdruckers.....	5-10
5.4.5	Löschen von Druckern oder Klassen.....	5-10
5.5	Der Scheduler.....	5-11
5.5.1	Starten des Schedulers durch lpsched.....	5-12
5.5.2	Stoppen des Schedulers durch lpshut.....	5-12
5.6	Schnittstellenprogramme für Drucker.....	5-13
5.6.1	Generierung von Kommandozeilen.....	5-14
5.6.2	Ausgabe mit Hilfe von Schnittstellenprogrammen.....	5-14

Inhaltsverzeichnis

5.6.3	Rückgabecodes der Schnittstellenprogramme.....	5-15
5.7	Einsatz von Geräten und Terminals als Drucker.....	5-16
5.7.1	Geräte.....	5-16
5.7.2	Terminals.....	5-17
A1	Anhang 1: Administrator-Kommandos.....	A1-1
A2	Anhang 2: Gerätedateien.....	A2-1
A3	Anhang 3: Dateiformate.....	A3-1
A4	Anhang 4: Stichwortverzeichnis.....	A4-1

Einleitung

1 Einleitung

Das vorliegende Handbuch dient als Arbeitsgrundlage und Nachschlagewerk und soll als solches den Systemverwalter bei seinen vielfältigen Aufgaben unterstützen.

Das Handbuch ist in fünf Sektionen eingeteilt:

1. Kurzbeschreibung der Hardware des Systems TARGON /31, Modell 30 und Modell 50.
2. Booting der TARGON /31. Dieses Kapitel beschreibt die einzelnen Bootphasen.
3. Laufende Arbeiten der Systemverwaltung. Diese Sektion enthält Anweisungen und Vorschläge, um einen sicheren Systembetrieb zu gewährleisten.
4. Beschreibung des Drucker-Spoolsystems.
5. Drei Anhänge, in denen Kommandos, Gerätedateien und Dateiformate beschrieben werden, die für den Systemadministrator von Bedeutung sind.

Die Notation der Syntax ist folgendermaßen aufgebaut:

kursiv gedruckte Worte stehen als Symbole, für die konkrete Werte oder Namen eingesetzt werden müssen,

[...] in eckigen Klammern eingeschlossene Teile können fehlen,

... drei Punkte bedeuten, daß das unmittelbar vorausgehende Element beliebig oft wiederholt werden kann.

Alle übrigen Zeichen und Worte in Normalschrift sind Endsymbole und müssen stets geschrieben werden.

Einleitung

System-Installation

Die Installation des Systems TARGON /31 wird durch Mitarbeiter der Nixdorf Computer AG durchgeführt. Deshalb wird in diesem Handbuch auf die Hardware und die jeweilige Aufgabe der Moduln nur kurz eingegangen.

System-Instandhaltung und -Ausbau

Die System-Instandhaltung, der System-Ausbau sowie Diagnosefunktionen werden ausschließlich von Nixdorf-Mitarbeitern durchgeführt. Sie besitzen spezielle Kenntnisse und Fertigkeiten, die einen problemlosen Systembetrieb, das Auswechseln von Logik-Moduln und den Anschluß von zusätzlicher Peripherie ermöglichen.

Kurzbeschreibung Hardware TARGON /31

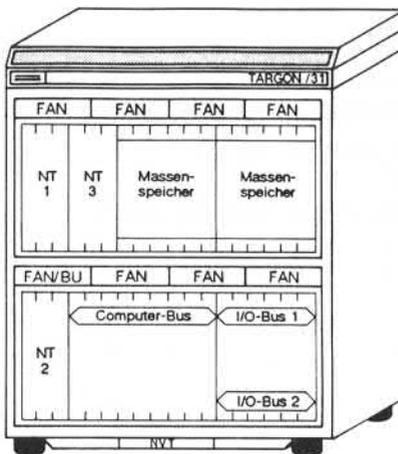
2 Kurzbeschreibung Hardware TARGON /31

In diesem Kapitel werden die HW-Komponenten für das System TARGON /31, Modell 30 und Modell 50, mit ihren Aufgaben und Eigenschaften im System beschrieben.

2.1 TARGON /31, Modell 30

Das Modell 30 ist wie folgt aufgebaut:

- EKK-Chassis, zwei Etagen, 18 Slot breit
- Untere Etage: 3 Slot Netzteil
9 Slot Computerrückwand
6 Slot I/O-Rückwand
- Obere Etage: 6 Slot für zwei Netzteile
12 Slot für zwei 5¼"-Massenspeicher-Chassis
- Anschlußfeld in der unteren Etage hinter der ZE-Rückwand zur Aufnahme der Anschlußkarten.



Kurzbeschreibung Hardware TARGON /31

2.1.1 CPU-Ausstattung, Modell 30

Die CPU-Ausstattung der Modelle 30 kann wie folgt konfiguriert werden:

Computerrückwand 9 Slot

Slot	Modul	Grundausrüstung
1	System-Peripherie-Prozessor (SPP)	Ja
2-3	Applikations-Prozessor (AP)	Ja
4-5	Memory, 4-8 MB	4 MB
6	Reserviert	-
7	Reserviert	-
8	I/O-Prozessor II	-
9	I/O-Prozessor I	Ja

I/O-Rückwand

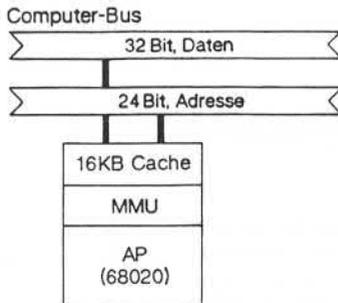
Slot	Modul	Grundausrüstung
1	Terminal-Communication-Controller (TCC)	Ja
2	Multi-Funktions-Controller (MFC) (DP und SAS II)	-
3-5	Terminal-Communication-Controller (TCC)	-
6	Reserviert	-

Kurzbeschreibung Hardware TARGON /31

2.1.2 Prozessorausstattung, Modell 30 und Modell 50

Die Modelle 30 und 50 sind in der Grundausstattung mit einem Applikations-Prozessor (AP) ausgestattet.

Der AP basiert auf dem Prozessor MC 68020 (16.6 MHz Taktrate), dem Hardware-Cache (16 KB) und der Memory-Management-Unit (MMU). (Übergangsweise basiert der AP noch auf dem Prozessor 68010 (12,5 MHz Taktrate und 8 KB Hardware-Cache.) Der AP ist ein Doppelschub und belegt 2 Slot der Computerrückwand.



Kurzbeschreibung Hardware TARGON /31

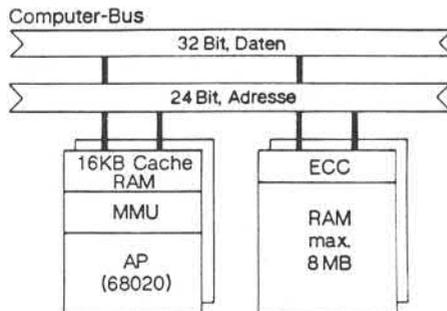
2.1.3 Speicherausstattung, Modell 30 und Modell 50

Die maximale Speicherausstattung ist derzeit 8 MB, zwei 4-MB-Einschübe. Ein Einschub mit 4 MB ist in der Grundausstattung enthalten.

Für die Speicherkonfigurierung sind Speichereinschübe mit der Kapazität 4 MB verfügbar. Die maximale Speicherausstattung ist derzeit auf 8 MB begrenzt.

Der Speicher wird durch folgende Funktionen gesichert:

- Parity-Prüfung
- Notstromversorgung
- Error-Correction-Control (ECC),
1 Bit Fehlerkorrektur und > 1 Bit Fehlererkennung



Kurzbeschreibung Hardware TARGON /31

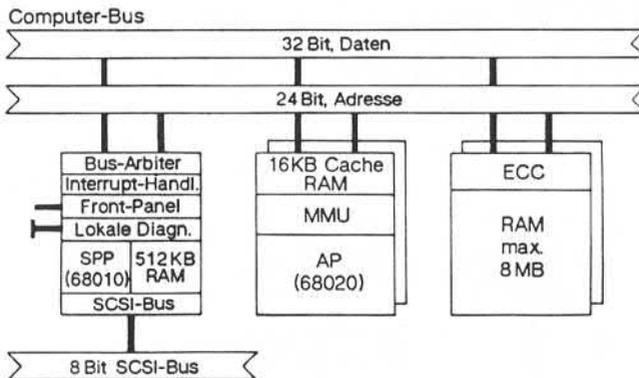
2.1.4 System-Peripherie-Prozessor (SPP)

Der SPP ist in der Grundausstattung des Modells 30 enthalten.

Der SPP hat im Modell 30 folgende Funktionalität:

- Busschiedsrichter
- Interruptverwaltung
- Steuerung der Bediener-Anzeige
- Lokales Diagnoseinterface (TKD)
- Steuerung der Massenspeichercontroller (SCSI-Bus)

Der SPP verfügt über einen 512 KB Internspeicher und einen Prozessor MC 68010.



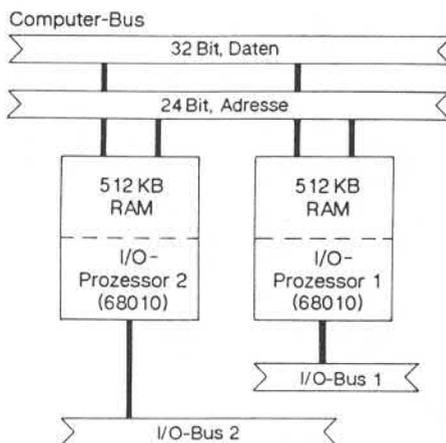
© Weitergabe sowie Vervielfältigung dieses Unterlagsmaterials, Vervielfältigung und Verbreitung, auch auszugsweise, ist ohne schriftliche Genehmigung der Nixdorf Computer AG. In Zweifelsfällen sind die Verantwortlichen zu Schadensersatz verpflichtet. Alle Rechte für den Fall der Patenterteilung oder Gebrauchsmustererteilung vorbehalten.

Kurzbeschreibung Hardware TARGON /31

2.1.5 Input/Output-Prozessor (IOP), Modell 30 und Modell 50

Zusätzlich zum AP stehen dedizierte IOPs für die Steuerung der I/O-Controller zur Verfügung. In den Modellen 30 und 50 können maximal zwei IOPs konfiguriert werden. Ein IOP ist bereits in der Grundausstattung der Modelle 30 und 50 enthalten. Der IOP entlastet den AP wesentlich von der zeitintensiven Zeichenverarbeitung im Interruptmodus. Über den IOP werden die zeichenorientierten Protokolle für den Terminaldialogbetrieb, die Druckerausgaben und für Datenfernübertragungszwecke abgewickelt.

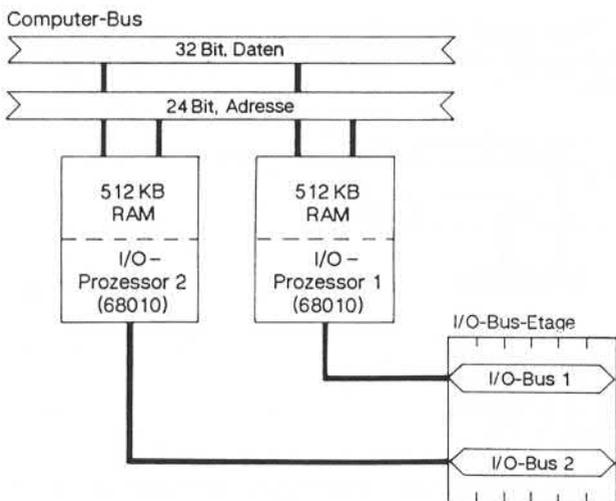
Jeder IOP verfügt über einen Internspeicher von 512 KB und einen Prozessor MC 68010. Pro IOP wird für den Anschluß der I/O-Controller jeweils ein I/O-Bus bereitgestellt.



Kurzbeschreibung Hardware TARGON /31

2.1.6 Input/Output-Controllerausstattung, Modell 30 und Modell 50

Maximal sind zwei I/O-Busse (2 x IOP) pro Modell 30 und 50 möglich.



Jeder I/O-Controller hat einen Dual-Port-Anschluß zur I/O-Etage. Dem jeweiligen IOP wird per Software-Einstellung mitgeteilt, welchen I/O-Controller er steuert.

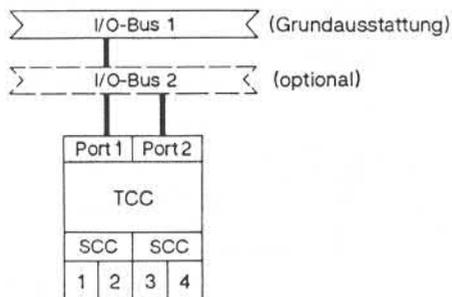
© „Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Mitteilung ihres Inhalts nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zu Schadenersatz. Alle Rechte für den Fall der Patenterteilung oder Gebrauchsmustererteilung vorbehalten.“

 Kurzbeschreibung Hardware TARGON /31

2.1.6.1 Terminal-Communication-Controller (TCC)

Der TCC kann serielle Anschlüsse (z. B. Bildschirmarbeitsplätze oder DFÜ-Leitungen) synchron oder asynchron betreiben.

Pro TCC stehen vier unabhängige Kanäle zur Verfügung, die derzeit paarweise mit unterschiedlichen Schnittstellenkarten bestückt werden können.

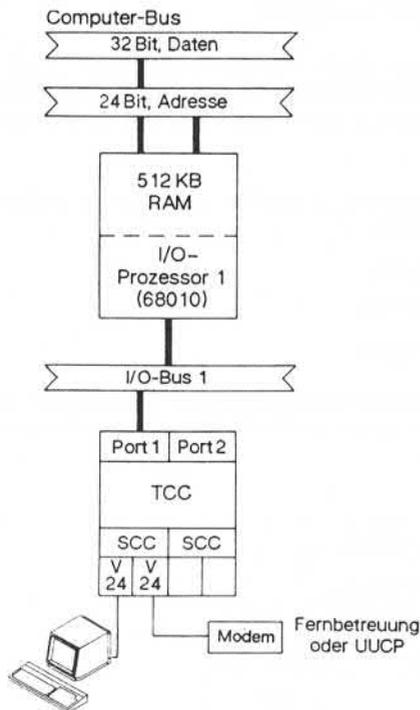


Folgende Schnittstellenkarten können eingesetzt werden:

- ☞ V.24/V.28 (RS232C)
- ☞ IHSS
- ☞ 20 mA
- ☞ V.11/X.21 (DATEX)

In der Grundausrüstung der Modelle 30 und 50 sind jeweils ein TCC mit zwei V.24/V.28-Schnittstellenkarten enthalten.

Kurzbeschreibung Hardware TARGON /31



An dem ersten Kanal wird die Systemkonsole angeschlossen. Der zweite Kanal ist für die Fernbetreuung und UNIX to UNIX Copy (UUCP) konfiguriert. In der TARGON /31 können maximal vier TCC's konfiguriert werden. Es stehen somit noch 14 Point-to-Point-Kanäle (16 minus 2 Kanäle wie oben beschrieben) zur Verfügung.

© Weitergabe sowie Vervielfältigung dieser Unterlagen, Vernetzung und Mitteilung ihres Inhalts nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zu Schadensersatz. Alle Rechte für den Fall der Patenterteilung oder Gebrauchsmustereintragung vorbehalten.

 Kurzbeschreibung Hardware TARGON /31

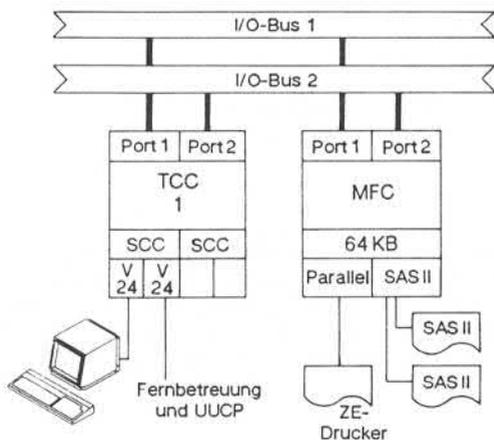
2.1.6.2 Der Multifunktions-Controller (MFC)

Über den MFC können Systemdrucker angesteuert werden. Der MFC unterstützt eine Parallel-Schnittstelle (NCAG-DP, Dataproducts-Shortline oder Dataproducts-Longline) und eine serielle Schnittstelle (SAS-Protokoll).

Die Parallel-Schnittstelle wird über Steckkarten betrieben. Folgende Steckkarten sind einsetzbar:

- Nixdorf Dataproducts
- Dataproducts-Shortline
- Dataproducts-Longline

Über die serielle Schnittstelle können zwei Drucker betrieben werden. Der MFC verfügt über einen 64 KB Puffer und wird durch einen 80188 Prozessor gesteuert.

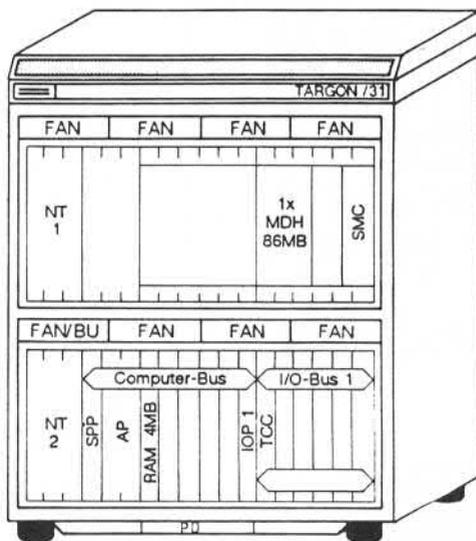


Kurzbeschreibung Hardware TARGON /31

2.1.7 Übersicht Modell 30

Die folgenden Abbildungen geben einen Überblick über die Hardware des Modells 30 in der Grund- und in der Maximalausstattung.

2.1.7.1 Grundausrüstung

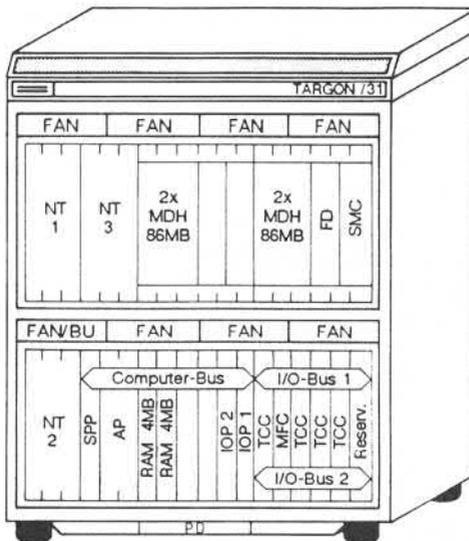


- NT = Netzteil
- PD = Power-Distributor
- MDH = Mini-Disk-High-Performance
- FD = Floppy-Disk
- SMC = Streaming-Mode-Cassette
- SPP = System-Peripherie-Prozessor
- AP = Applikations-Prozessor
- TCC = Terminal-Communication-Controller
- BU = Battery-Unit
- MFC = Multi-Funktions-Controller
- FAN = Lüfter

„Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Mitteilung ihres Inhalts nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlung wird strafrechtlich verfolgt. Die Haftung für Druck- oder Patentfehler oder Gebrauchsmusteränderung vorbehalten.“

 Kurzbeschreibung Hardware TARGON /31

2.1.7.2 Maximalausstattung



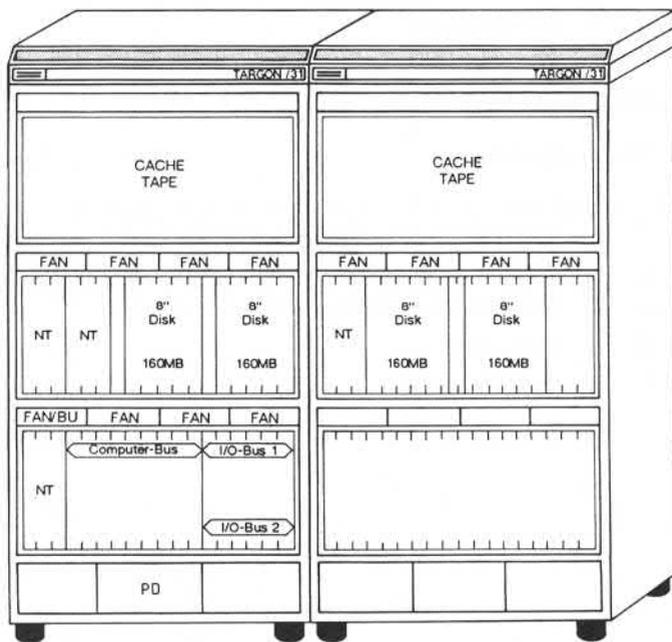
- NT = Netzteil
 PD = Power-Distributor
 MDH = Mini-Disk-High-Performance
 FD = Floppy-Disk
 SMC = Streaming-Mode-Cassette
 SPP = System-Peripherie-Prozessor
 AP = Applikations-Prozessor
 TCC = Terminal-Communication-Controller
 BU = Battery-Unit
 MFC = Multi-Funktions-Controller
 FAN = Lüfter

Kurzbeschreibung Hardware TARGON /31

2.2 TARGON /31, Modell 50

Das Modell 50 ist wie folgt aufgebaut:

- EKK-Chassis, drei Etagen, 18 Slot breit.
- Untere Etage: 3 Slot Netzteil
9 Slot Computerrückwand
6 Slot I/O-Rückwand
- Mittlere Etage: 2 x 3 Slot Netzteil
12 Slot für zwei 8"-Magnetplatten
- Obere Etage: Magnetbandgerät
- Anschlußfeld in der unteren Etage hinter der ZE-Rückwand.
- Optionaler Peripherieschrank zur Erweiterung von Magnetplatten- oder Magnetband-Peripherie.



© „Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Mitteilung ihres Inhalts nicht gestattet, soweit nicht ausdrücklich zugestanden. Insbesondere ist das Kopieren von Texten, Bildern und sonstigen Angaben ohne schriftliche Genehmigung vorbehalten.“

 Kurzbeschreibung Hardware TARGON /31

2.2.1 CPU-Ausstattung, Modell 50

Die CPU-Ausstattung des Modells 50 kann wie folgt konfiguriert werden:

Computerrückwand 9 Slot

Slot	Modul	Grundausstattung
1	System-Peripherie-Prozessor (SPP)	Ja
2-3	Applikations-Prozessor (AP)	Ja
4-5	Memory, 4-8 MB	4 MB
6-7	Disk/Tape-Prozessor (DTP)	Ja
8	I/O-Prozessor II	-
9	I/O-Prozessor I	Ja

I/O-Rückwand

Slot	Modul	Grundausstattung
1	Terminal-Communication-Controller (TCC)	Ja
2	Multi-Funktions-Controller (MFC)	-
3-5	Terminal-Communication-Controller (TCC)	-
6	Reserviert	-

Kurzbeschreibung Hardware TARGON /31

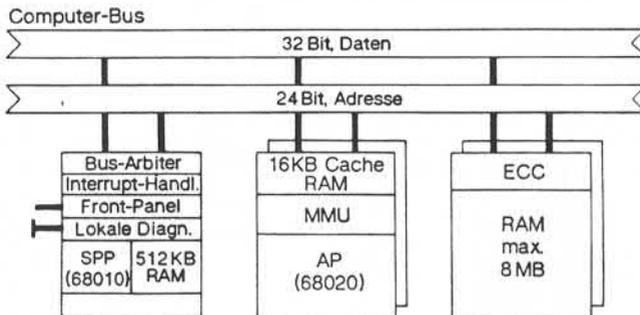
2.2.2 System-Peripherie-Prozessor (SPP)

Der SPP ist in der Grundausstattung des Modells 50 enthalten.

Der SPP hat im Modell 50 folgende Funktionalität:

- Busschiedsrichter
- Interruptverwaltung
- Steuerung der Bediener-Anzeige
- Lokales Diagnoseinterface (TKD)

Der SPP verfügt über einen 512 KB Internspeicher und einen Prozessor MC 68010.



© „Weitergabe sowie Vervielfältigung dieser Unterlage, Verbreitung und sonstiger Gebrauch ohne schriftliche Genehmigung von Nixdorf Computer AG. Die Weitergabe und die Vervielfältigung dieser Unterlage ist, soweit nicht ausdrücklich anders bestimmt, ausdrücklich untersagt. Alle Rechte für den Fall der Patenterteilung oder Gebrauchsmustereintragung vorbehalten.“

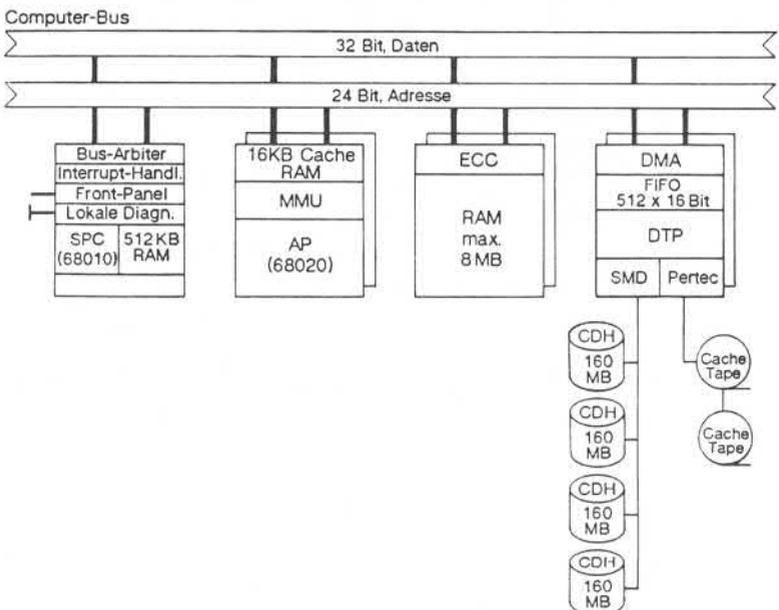
Kurzbeschreibung Hardware TARGON /31

2.2.3 Disk/Tape-Prozessor, Modell 50 (DTP)

Im Gegensatz zum Modell 30, wo die Massenspeicherperipherie über den SCSI-Bus gesteuert wird, verfügt das Modell 50 über einen Disk/Tape-Prozessor (Grundausrüstung). Der DTP ist direkt am Computer-Bus angeschlossen.

Der DTP ist ein Bit-Slice-Prozessor und hat folgende Funktionalität:

- DMA zum Computer-Bus
- Interner Puffer für Disk und Tape
- Multicommand-Interface
- Paralleles Suchen (Disk)
- Vier Disk-Kanäle (SMD), ein Tape-Kanal (Pertec)
- Burst-Error-Prozessor (korrigiert 11 Bit pro Sektor)

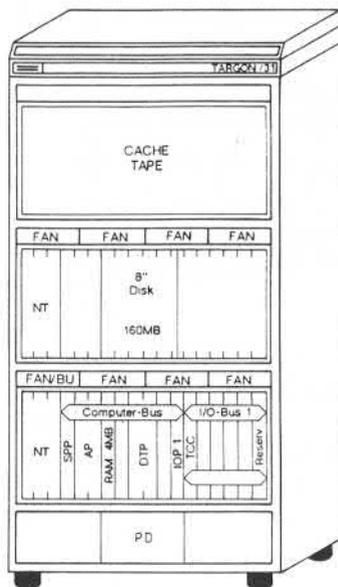


Kurzbeschreibung Hardware TARGON /31

2.2.4 Übersicht Modell 50

Die folgenden Abbildungen geben einen Überblick über die Hardware des Modells 50 in der Grund- und in der Maximalausstattung.

2.2.4.1 Grundausrüstung

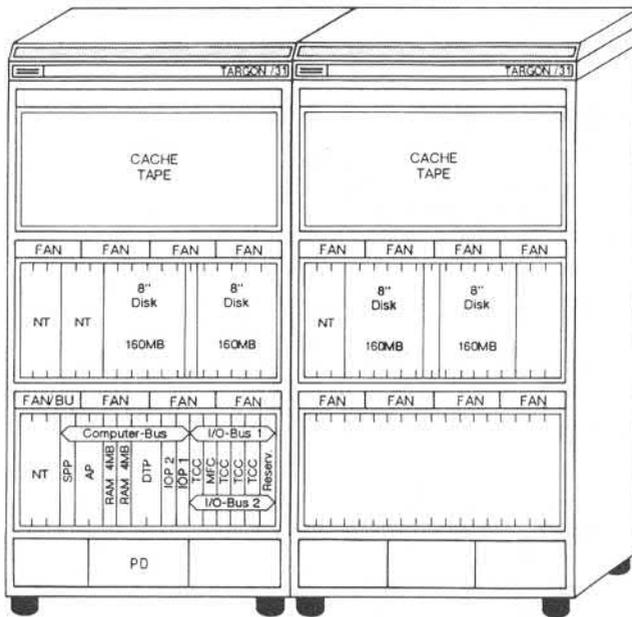


- NT = Netzteil
- PD = Power-Distributor
- MDH = Mini-Disk-High-Performance
- FD = Floppy-Disk
- SMC = Streaming-Mode-Cassette
- SPP = System-Peripherie-Prozessor
- AP = Applikations-Prozessor
- TCC = Terminal-Communication-Controller
- BU = Battery-Unit
- MFC = Multi-Funktions-Controller
- DTP = Disk/Tape-Prozessor
- FAN = Lüfter

© Weitergabe sowie Vervielfältigung dieser Unterlagen, Vervielfältigung und Verbreitung, auch auszugsweise, ist ohne schriftliche Genehmigung der Nixdorf Computer AG. Die Nixdorf Computer AG übernimmt keine Haftung für die Richtigkeit der Angaben. Alle Rechte für den Fall der Patenterteilung oder Gebrauchsmustereintragung vorbehalten.

 Kurzbeschreibung Hardware TARGON /31

2.2.4.2 Maximalausstattung



- NT = Netzteil
 PD = Power-Distributor
 MDH = Mini-Disk-High-Performance
 FD = Floppy-Disk
 SMC = Streaming-Mode-Cassette
 SPP = System-Peripherie-Prozessor
 AP = Applikations-Prozessor
 TCC = Terminal-Communication-Controller
 BU = Battery-Unit
 MFC = Multi-Funktions-Controller
 DTP = Disk/Tape-Prozessor
 FAN = Lüfter

Kurzbeschreibung Hardware TARGON /31

2.3 Peripherie

An die Systeme TARGON /31 ist eine breite Palette von Peripheriegeräten anschließbar:

- Magnetplattenlaufwerke
 - Winchester-Festplatten mit 80 Megabyte (Modell 30) und 160 Megabyte (Modell 50)
 - Magnetbandgeräte
 - Streamer-Cassette mit 8000 bpi und 90 ips (Modell 30)
 - Cache-Streamer-Tape mit 1600/3200 bpi und 100/50 ips (Modell 50)
 - Drucker
 - Zeilendrucker mit 600 Zeilen/Minute
 - Matrixzeilendrucker mit 300 Zeilen/Minute
 - serielle Drucker 75/150/210 Zeichen/Sekunde
 - Terminals
 - Zeichenorientierte Terminals
 - Positivdarstellung
 - Terminalperipherie (Drucker)
 - Kommunikation
 - Programm-Programm-Kommunikation (MSV 2)
 - RJE/BSC 2780/3780
-

Booten der TARGON /31

3 Booten der TARGON /31

Das Boot-Verfahren ist in mehrere Stufen unterteilt, um eine größtmögliche Sicherheit und Flexibilität zu erreichen. In diesem Kapitel finden Sie eine Beschreibung der verschiedenen Boot-Phasen der TARGON /31.

3.1 Boot-Phasen

Auf jedem Prozessor-Board befindet sich ein PROM, der die folgenden Programme enthält:

- Startsmail (Hardware-Testprogramme) und
- First Level Boot (Boot1).

Der First Level Boot lädt von Platte, Band oder Floppy die letzte Ladestufe, den

- Second Level Boot (Boot2).

3.2 Startsmail

Ist der System-Hauptschalter eingeschaltet (Normalfall), wird das System mit dem rechten Schlüssel am Frontpanel eingeschaltet und in einen definierten Anfangszustand gebracht (RESET). Dazu müssen Sie den Schlüssel nach rechts in Stellung 1 drehen. Nach dem Loslassen geht er automatisch in seine Ruhestellung zurück. Soll erneut ein RESET erfolgen, genügt es den Schlüssel in Stellung 1 zu bringen, wobei das System nicht ausgeschaltet wird. Soll das System jedoch ausgeschaltet werden, drehen Sie den Schlüssel links herum in die Stellung 0. Nach dem Loslassen springt der Schlüssel in die Ruhestellung zurück.

Booten der TARGON /31

Nach einem RESET wird der Hardware-Selbsttest (Startsmal) und danach die erste Phase des Bootstrap gestartet. Der weitere Ablauf des Bootens ist abhängig von der Stellung des Auswahlschalters (linker Schlüssel).

Beim Restart des Systems wird also als erste Aktion der sogenannte Startsmal aktiviert. Diese Testsoftware prüft alle Boardfunktionen sowie die Verbindung der Prozessoren untereinander. Den Verlauf des Startsmal können Sie auf der LED-Anzeige des Bedienungspanels verfolgen:

- 00 Startsmal gestartet.
- 01 Applikations-Prozessor wurde erkannt.
- 03 Warten auf Antwort vom Applikations-Prozessor.
- 04 Testen internes RAM.
- 06 Testen der Parity-Logik.
- 08 Initialisieren der Diagnose-Register des ersten MEM-Boards.
- 0A Siehe 0C.
- 0C Der Diagnose-Prozessor wird in Test-Mode versetzt.
- 0E Hauptspeichertest des ersten MB. Der restliche Hauptspeicher wird durch die anderen Boards getestet.
- 10 Prüfen der EDEC-Logik des Hauptspeichers.
- 12 Hauptspeicherinitialisierung. Verlauf wie bei Test des Hauptspeichers.
- 14 Testen der Hauptspeicher-Adreßlogik. Testen des Adreß-Busses auf Fehler durch Einschreiben von definierten Mustern.
- 16 Siehe 18.
- 18 MMU-Test im User-Mode. Test erfolgt durch den Applikations-Prozessor.
- 1F Ende von Startsmal.

Ist der Startsmal fehlerfrei beendet, kann das eigentliche Laden des Systems beginnen.

Booten der TARGON /31

3.3 First Level Boot

Der Boot-Prozeß selbst ist in zwei Teile geteilt, den PROM-residenten **First Level Boot** (Ladestufe 1), der den **Second Level Boot** (Ladestufe 2) von einer Platte, einem Band oder einer Floppy-Disk in den Hauptspeicher lädt.

Der First Level Boot arbeitet auf den Modellen 30 und 50 abhängig von der Stellung des linken Schlüssels am Frontpanel.

Schlüsselstellung	Model 30	Model 50	M30 und M50
1		Standalone	
2		Manual Boot	
3		Autoboot	
5	Standalone		
6	Manual Boot		
7	Autoboot		
9			Dump Image

Booten der TARGON /31

3.3.1 Autoboot

Die Schlüsselstellung für Autoboot sollte für den Normalbetrieb gewählt werden. Hierbei lädt der First Level Boot den Second Level Boot von der Platte, unterdrückt die Ausgabe des Menüs vom Second Level Boot und lädt eine Standardversion des Betriebssystems in den Hauptspeicher. Bedienereingaben werden beim Autoboot nicht verlangt.

Die Standard-Betriebssystemversion wird in das Verzeichnis /os geladen und besteht aus den folgenden drei Teilen:

- spc (System Peripherals Controller),
- unix und
- iop (I/O-Prozessor).

Hinweis: Wollen Sie eine andere Betriebsversion laden, müssen Sie die Schlüsselstellung für den Manual Boot wählen.

Nach dem Laden der drei Betriebssystemteile erscheint die folgende Nachricht auf der Konsole:

ENTER RUN LEVEL (0-6, s or S)

Geben Sie **s** oder **S** ein, befindet sich Ihre TARGON /31 im Single-User-Modus; die Eingabe einer Ziffer bewirkt das Wechseln in den Multi-User-Betrieb.

Booten der TARGON /31

3.3.2 Standalone

Diese Schlüsselstellung benötigen Sie, um Standalone-Programme zu laden, z. B. um eine neue Anlage zu installieren. Beim Installieren der Maschine wird davon ausgegangen, daß kein Dateisystem vorhanden ist. Die benötigte Software für die TARGON /31 wird entweder auf einem Magnetband (Modell 50) oder auf einer Streamer-Cassette (Modell 30) ausgeliefert. Diese Speichermedien enthalten jeweils:

- Standalone Second Level Boot und
- Root-Dateisystem.

Der Second Level Boot befindet sich jeweils auf einem gesonderten Band.

Zum Laden des Standalone Second Level Boot gehen Sie folgendermaßen vor:

1. Legen Sie das Speichermedium (Band oder Cassette) in das entsprechende Laufwerk ein.
2. Vergewissern Sie sich, daß der linke Schlüssel am Frontpanel in der richtigen Stellung für Standalone ist. Drehen Sie anschließend den rechten Schlüssel nach rechts in die Stellung 1. Nach dem Loslassen geht er automatisch in seine Ruhestellung zurück. Jetzt läuft der Hardware-Selbsttest Startsmall.
3. Nach Beendigung des Startsmall (ca. eine Minute) wird nun der Second Level Boot vom entsprechenden Speichermedium eingelesen.

Danach ist der Second Level Boot im Hauptspeicher und das Menü wird angezeigt. Die Arbeitsweise der im Menü angebotenen Funktionen finden Sie im Kapitel „Second Level Boot“.

Jetzt können Sie das Root-Dateisystem auf die Platte bringen. Vergewissern Sie sich, daß das richtige Band im Laufwerk ist. Die Auswahl **t** startet den Kopiervorgang. (Nähere Einzelheiten entnehmen Sie bitte dem Kapitel „Einlesen des Root-Dateisystems“.

Booten der TARGON /31

Ist dieser Vorgang abgeschlossen, wird wieder das Auswahlmenü des Second Level Boot eingeblendet. Durch Eingabe von **b** und **CR** wird das Betriebssystem geladen (siehe auch Kapitel „Booten des Betriebssystems“).

Hinweis: Die Installationssoftware sollten Sie aufbewahren, damit sie im Notfall wieder benutzt werden kann.

3.3.3 Manual Boot

Bei der Wahl dieser Schlüsselstellung wird der Second Level Boot von der Platte gelesen, anschließend wird in das Auswahlmenü verzweigt.

Die Arbeitsweise der im Menü angebotenen Funktionen finden Sie im Kapitel „Second Level Boot“.

3.3.4 Dump Image

Ein Speicherabbild wird auf Band ausgelagert. Diese Funktion sollten Sie nach Systemabstürzen anwählen. Legen Sie ein Band bzw. eine Cassette in das Laufwerk, bevor Sie die Schlüsselstellung 9 einstellen. Drehen Sie anschließend den rechten Schlüssel auf 1 und lassen ihn wieder los. Nach einem Reboot können Sie das Band mit

```
dd if=/dev/rmt0 of=dumpfile bs=2k
```

wieder einlesen. Mit Hilfe des Administrator-Kommandos `crash` kann dann der Inhalt von `dumpfile` analysiert werden.

Booten der TARGON /31

3

3.4 Second Level Boot

Wie in den vorangegangenen Kapiteln beschrieben, wird Ihnen bei den Schlüsselstellungen „Manual Boot“ und „Standalone“ das folgende Auswahlmü des Second Level Boot angezeigt:

Nixdorf TARGON /31 - Model xx 2nd Level boot from drive : x

- (b) .. Boot OS from disk**
- (d) .. Dump root file system to tape**
- (t) .. Load tape to root file system**
- (s) .. Set drive number**
- (k) .. Write 2nd level boot to tape**
- (f) .. Format disk**
- (e) .. Execute file**
- (q) .. Run diagnostic programs**
- (h) .. (Hardware) testprograms**

Die jeweilige Funktion kann durch Eingabe des Kennbuchstabens ausgewählt werden. Werden noch Parameter benötigt, werden diese vom Programm interaktiv angefordert.

In den folgenden Abschnitten finden Sie nähere Erläuterungen zu den einzelnen Unterfunktionen des Second Level Boot, die vom Systemadministrator genutzt werden können. Die Funktionen e, q und h sind für die Mitarbeiter des Technischen Kundendienstes reserviert und sollten von Ihnen nicht ausgewählt werden.

Booten der TARGON /31

3.4.1 Booten des Betriebssystems

Durch die Eingabe von **b** (Boot OS from disk) wird das Laden des UNIX-Betriebssystems eingeleitet. Die Meldung

Starting Job, Spc, Iop

zeigt, daß der Bootvorgang erfolgreich verläuft.

Nach den Meldungen über Releasestand, Version usw. meldet sich der Init-Prozeß, der den Start des restlichen Systems steuert.

Optional kann auch eine Version angegeben werden, so daß der Start eines neugenerierten Kernels möglich ist. Haben Sie z. B. eine neue Version der Standardversionen von spc, unix und iop mit der Namensendung .exp erstellt, brauchen Sie nur die Namensendung – in unserem Beispiel exp – anzugeben und diese Version des Betriebssystems wird geladen. Im Fehlerfall erhalten Sie folgende Nachricht:

Unable to load file xxxxxxxx

In diesem Zusammenhang verweisen wir auf die Systemliteratur TARGON /31 „Systemgenerierung und Konfigurierung“. In diesem Handbuch finden Sie genaue Hinweise zur Erstellung eines neuen UNIX-Kerns.

Booten der TARGON /31

3.4.2 Sichern des Root-Dateisystems

Nach der Installation des Systems und der Anpassung an spezielle Bedürfnisse (inittab, passwd, gettydefs usw.) sollten Sie das Root-Dateisystem sichern (Funktion: Dump root file system to tape). Dazu müssen Sie – je nach Modell – entweder eine Streamer Cassette oder ein Magnetband in das entsprechende Laufwerk einlegen. Beim Modell 50 wird automatisch auf HIGH Density (3200 bpi) umgeschaltet. Nach Durchführung der Sicherung erfolgt das Zurücksetzen auf LOW DENSITY (1600 bpi).

Die Größe des Root-Dateisystems wird automatisch festgestellt, muß jedoch sicherheitshalber vom Operator bestätigt werden. Die Größe beträgt standardmäßig 4000 Blöcke.

Die anschließende Frage

Backup via BACKUP/RESTORE? :

beantworten Sie mit **y**. Die Beendigung der Sicherung, die beim Modell 30 ca. fünf Minuten dauert, wird durch folgende Nachricht angezeigt:

Disk to Tape copy done - Rewind in progress

Booten der TARGON /31

3.4.3 Einlesen des Root-Dateisystems

Das Root-Dateisystem wird vom Speichermedium auf die Platte geschrieben (Funktion: Load tape to root file system). Ist das Band ordnungsgemäß im Laufwerk untergebracht, erscheint die folgende Meldung:

Copy xxx blocks to disk drive yyy OK (y/n):

Die angezeigte Anzahl Blöcke beträgt 4000.

Nachdem Sie die Anfrage mit **y** bestätigt haben, beginnt der Kopiervorgang. Die Frage

Restore via BACKUP/RESTORE

muß mit **y** beantwortet werden, um ein schnelles Kopieren anzustoßen. Nach ca. fünf Minuten ist der Kopiervorgang beendet und die folgende Meldung wird ausgegeben:

Load of Tape to disk complete - Rewind in progress

Bei einem Modell 50 wird das Kopieren folgendermaßen vorgenommen:

1. Das Band wird auf HIGH Density (3200 bpi) gesetzt.
2. Die Blöcke werden vom Band gelesen und auf die Platte geschrieben.
3. Das Kopierprogramm unterrichtet Sie über seine jeweiligen Aktivitäten durch Meldungen.

Nach ca. vier Minuten werden Sie über die Beendigung des Kopierens durch folgende Nachricht

Load to tape from disk - Rewind in progress

unterrichtet. Anschließend wird das Band auf LOW DENSITY (1600 bpi) zurückgesetzt.

Die Sicherung beinhaltet die in den nächsten beiden Kapiteln aufgeführten Verzeichnisse des Root- und /usr-Dateisystems.

Booten der TARGON /31

3.4.3.1 Inhalt des Root-Dateisystems

Das Root-Dateisystem enthält folgende Verzeichnisse:

- **bin**
Benutzerkommandos
- **dev**
Gerätedateien
- **etc**
Administrator-Kommandos und Systemdateien
- **lib**
Bibliotheksfunktionen (Assembler, C usw.)
- **lost+found**
Wird vom Dateiprüfprogramm fsck benutzt, um I-Knoten, die aus dem Dateisystem entfernt wurden, zu retten.
- **usr**
Verzeichnis zum Einhängen des Dateisystems /usr
- **tmp**
Temporärdateien. Diese Dateien müssen bei jedem Systemstart gelöscht werden, da sie schnell anwachsen.

Booten der TARGON /31

3.4.3.2 Inhalt des Dateisystems /usr

Das Dateisystem /usr enthält die folgenden Verzeichnisse:

- **adm**
Administrator-Kommandos und Dateien, die Accounting-Informationen enthalten.
- **bin**
Benutzerkommandos
- **include**
C-include-Dateien
- **lib**
Archivbibliotheken, z. B. für Textverarbeitung
- **mail**
Mail-Dateien
- **lost+found**
Wird vom Dateiprüfprogramm fsck benutzt, um I-Knoten, die aus dem Dateisystem entfernt wurden, zu retten.
- **news**
Verzeichnis für die Systemnachrichten
- **pub**
Informationsdateien für die Benutzer, z. B. ASCII-Code-Tabelle
- **release**
Freigabedokumentation
- **spool**
Dämonprozesse
- **tmp**
Temporärdateien. Diese Dateien müssen bei jedem Systemstart gelöscht werden, da sie schnell anwachsen.

Booten der TARGON /31



3.4.4 Auswahl des Boot-Laufwerks

Die Laufwerksnummer, von der das Root-System gebootet werden soll, kann hier angegeben werden (Funktion: Set drive number). Dadurch ist es möglich, die Anlage von verschiedenen Plattenlaufwerken zu laden.

3.4.5 Sichern des Second Level Boot

Durch Eingabe der Option **k**, die Ihnen im Menü des Second Level Boot angeboten wird, können Sie das Boot2-Programm auf Band schreiben (Funktion: Write 2nd level boot to tape).

3.4.6 Platten formatieren

Nachdem Sie die Platte, die formatiert werden soll, dem System bekanntgemacht haben, werden Sie aus Sicherheitsgründen nach einem Paßwort gefragt (Funktion: Format Disk). Bei der TARGON /31 heißt dieses Paßwort **LOWEND**.

Durch das jetzt eingeblendete Menü wird der Typ der zu formatierenden Platte abgefragt. Bevor die Formatierung endgültig anläuft, werden Sie noch einmal gebeten, Ihre Eingaben zu bestätigen. Der Formatiervorgang dauert ca. 20 bis 40 Minuten (je nach Modell). Durch Meldungen werden Sie über die jeweiligen Aktivitäten des Programms unterrichtet.

Nach Beendigung der Formatierung erscheint die Meldung

- format completed -

gefolgt von Informationen über die Ersatzspur-Tabelle.

Booten der TARGON /31

Diese Ersatzspur-Tabellen werden für die Verwaltung inkonsistenter Blöcke benutzt. Sollte die Platte nicht 100 % in Ordnung sein, unterrichtet Sie der Plattenformatierer über die entsprechenden Spuruweisungen.

Hinweis: Zu viele inkonsistente Blöcke auf einer Platte beeinträchtigen die Performance des Systems erheblich.

Systemverwaltung TARGON /31

4 Systemverwaltung

Dieses Kapitel enthält die Beschreibung der Arbeiten, die im laufenden Betrieb erforderlich sind, und Informationen zu den Werkzeugen, die Sie benötigen, um diese Aufgaben zu erfüllen.

4

4.1 Verwaltungsdateien

Auf allen UNIX- und UNIX-kompatiblen Systemen enthält das Verzeichnis /etc u. a. eine Reihe von Dateien, die die Systemumgebung des Rechners beschreiben. Die für Sie als Systemadministrator wichtigen Dateien sind nachfolgend kurz beschrieben. Nähere Einzelheiten über diese Dateien entnehmen Sie bitte den entsprechenden Anhängen.

- /etc/inittab**
 Enthält Informationen über die von **init** zu startenden Prozesse.
- /etc/gettydefs**
 Enthält leitungsspezifische Parameter. Diese Datei wird von **getty** ausgewertet.
- /etc/rc**
 Diese Datei ist eine Shell-Prozedur und wird beim Wechsel vom Single-User-Modus in den Multi-User-Modus aufgerufen. Sie enthält Kommandos, die bei diesem Wechsel auszuführen sind, z. B. das Einhängen (mount) von Dateisystemen und den Start der Daemons. Daemons sind Prozesse, die ständig im Hintergrund ablaufen (z. B. cron). Die Ausgaben von /etc/rc erfolgen auf der Systemkonsole.
- /etc/passwd**
 Enthält für jeden zugangsberechtigten Benutzer eine Zeile mit benutzerspezifischen Informationen.
- /etc/group**
 Enthält die Gruppenzuordnung der Benutzer.
- /etc/profile**
 Diese Datei ist eine Shell-Prozedur und dient der Definition einer Standard-Benutzerumgebung.

Systemverwaltung TARGON /31

- **/etc/mnttab**
Wird beim Einhängen (mount) von Dateisystemen angelegt und ausgewertet.
- **/etc/checklist**
Wird von fsck ausgewertet; enthält die Namen der zu prüfenden Dateisysteme.
- **/etc/motd**
Enthält eine oder mehrere Tagesnachrichten (**message of the day**), die den Benutzern nach dem Anmelden angezeigt werden.
- **/etc/filesave** und **/etc/tapesave**
Diese Datei enthält Musterprozeduren für die lokale Datensicherung. Diese Datei sollte so modifiziert werden, daß die Prozeduren den lokalen Anforderungen gerecht werden.
- **/etc/shutdown**
Definition von Prozeduren zum ordentlichen Abschalten des Systems.

4.2 Eintragen neuer Benutzer

Es ist sehr einfach, das System TARGON /31 für neue Benutzer zugänglich zu machen. Folgende Informationen sollten Sie vorher einholen:

- Name des Benutzers (kann zur besseren Übersicht in das Feld für die Accounting-Information eingetragen werden).
- Login-Name (nicht mehr als acht Buchstaben, beginnend mit einem Kleinbuchstaben).
- Beziehung zu anderen Benutzern (fließt in die Auswahl der Gruppenzugehörigkeit ein).
- Grobe Abschätzung des erforderlichen Dateiraums (fließt in die Auswahl des Dateisystems ein).

Aufgrund dieser Informationen können Sie die passenden Einträge in die Dateien `/etc/passwd` und `/etc/group` vornehmen.

Systemverwaltung TARGON /31

Nachfolgend sind die einzelnen Schritte aufgezeigt, um einen neuen Benutzer zum System zuzulassen:

1. Anlegen eines Eintrags in `/etc/passwd` für den neuen Benutzer.
2. Editieren eines Eintrags in `/etc/group`.
3. Anlegen des in `/etc/passwd` eingetragenen Login-Verzeichnisses.
4. Ändern der Benutzer- und Gruppen-ID des soeben angelegten Login-Verzeichnisses für den neuen Benutzer (`chown`, `chgrp`).
5. Melden Sie sich zur Kontrolle unter dem neu angelegten Benutzer-namen an.
6. Eingeben eines Paßwortes für den neuen Benutzer.
7. Dem neuen Benutzer Login-Name und Paßwort mitteilen.

Der neue Benutzer kann nun mit dem System arbeiten. Er sollte sich dann ein selbstgewähltes Paßwort setzen (`passwd`).

Mit dem Kommando `newgrp` kann ein Benutzer in eine andere Gruppe wechseln. Voraussetzung ist jedoch, daß für ihn in der Datei `/etc/group` die Zugehörigkeit zu dieser anderen Gruppe eingetragen ist. Die Feststellung seiner Zugriffsrechte wird dann mit der neuen Gruppen-ID durchgeführt.

4.3

Datenschutzmechanismen

Einem Benutzer kann zur Ausführungszeit von Programmen der kontrollierte Zugriff auf Dateien ermöglicht werden. Dazu ist in den Zugriffsrechten des Programms das Set-User-ID- oder Set-Group-ID-Bit zu setzen (siehe Benutzerkommando `chmod`). Shell-Prozeduren ignorieren diese Funktion.

Systemverwaltung TARGON /31

Erkennbar sind diese Programme mit dem Kommando ls -l:

```
-rws rws rwx
```

Bei Ausführung erhält der Benutzer die Rechte der Gruppe des Programmeigentümers

Bei Ausführung erhält der Benutzer die Rechte des Programmeigentümers

Beispiel:

```
ls -l /bin/passwd
```

```
-rwsr-xr-x 1 root ... /bin/passwd
```

Programme, für die das Set-User-ID-Bit gesetzt ist, sollten Sie regelmäßig überprüfen. Das folgende Kommando erzeugt eine mail an root, die eine Liste aller root-eigenen Set-User-ID-Programme enthält.

```
find / -user root -perm -4100 -exec ls -l {} \; | mail root
```

Jede Veränderung in der mail sollte gründlich untersucht werden.

Noch einige Tips:

- Ändern Sie das Superuser-Paßwort regelmäßig. Nehmen Sie keine klaren eindeutigen Paßwörter, sondern wählen Sie eine Kombination aus sechs bis acht Zeichen, die aus alphanumerischen und Sonderzeichen besteht und keinen offensichtlichen Sinn ergibt.
- Wenn ein Wählanschluß (Fernübertragung) eingerichtet wird, sollte auch dieser Systemzugang unbedingt mit einem Paßwort gesichert werden.
- Die Kommandos chroot und su könnten mißbraucht werden. Es ist deshalb zu überlegen, ob diese Kommandos unbedingt benötigt werden oder ob es besser ist, sie aus dem System zu entfernen.

Systemverwaltung TARGON /31

4.4 Sicherung und Wiederherstellung von Daten

Es ist eminent wichtig, daß Daten nach Fehlern wieder rekonstruiert werden können. Dazu benötigen Sie in der Regel eine Sicherungskopie, die bei der Wiederherstellung Verwendung findet. In den folgenden beiden Kapiteln finden Sie Hinweise, wie Sie bei der Sicherung des Datenbestandes vorgehen sollten.

4.4.1 Schutzmechanismen für Benutzerdateien

Bei Systemzusammenbrüchen können eröffnete Dateien manchmal verloren gehen. Es ist dagegen sehr selten der Fall, daß ein komplettes Dateisystem zerstört wird. Nachfolgend finden Sie einen Vorschlag, wie Sie bei der Sicherung Ihres Datenbestandes vorgehen können:

- **Tägliche Sicherung:**
Jeden Tag sollten Sie eine inkrementelle Sicherung der Benutzer-Dateisysteme auf ein Sicherungsmedium durchführen. Diese Medien sollten bis zur nächsten Gesamtsicherung aufbewahrt werden, bevor sie erneut zur Sicherung benutzt werden. Als Sicherungsmedium kann ein Magnetband oder eine Streamer Cassette fungieren.
- **Wöchentliche Sicherung:**
Einmal pro Woche sollte jedes Dateisystem auf Magnetband kopiert werden. Diese Magnetbänder sollten Sie acht Wochen aufbewahren.
- **8-wöchige Sicherung:**
Diese Sicherungsbänder sollten für immer aufbewahrt werden. Jedes Jahr sollte ein Wiederkopieren erfolgen, d. h. die vom Band gelesene Information sollte erneut aufgeschrieben werden.

Sicherungen sollten nur im Single-User-Mode erfolgen. Außerdem ist es ratsam, vor einem Sicherungslauf das Dateisystem mit fsck zu überprüfen. Nur dann ist gewährleistet, daß die Sicherung fehlerfrei erfolgt.

Ist das Dateisystem einmal fehlerhaft, sollte eine Reparatur mit fsck versucht werden. Ist das Dateisystem zerstört, muß die zuletzt angefertigte komplette physikalische Sicherung eingelesen werden.

Systemverwaltung TARGON /31

4.4.2 Programme zur Sicherung von Dateisystemen

■ **find -cpio**

Die Option `-cpio` des `find`-Kommandos kann zur Sicherung von Dateien benutzt werden, die innerhalb einer definierten Periode verändert oder kreiert wurden.

■ **tar**

Tar operiert mit Dateinamen und sollte dann benutzt werden, wenn oft einzelne Dateien wieder eingespielt werden müssen oder wenn Sie Daten mit einem anderen UNIX-System austauschen wollen. Hierbei müssen Sie beachten, daß die Sicherung mit `tar` zwar schneller ist als die Sicherung mit `cpio`, `tar` sichert jedoch keine Gerätedateien und keine Named Pipes.

Nähere Beschreibungen des `find`- und `tar`-Kommandos finden Sie im Handbuch „Benutzerkommandos“.

Systemverwaltung TARGON /31

4.5 Hilfreiche Programme für den Systemverwalter

4.5.1 Kommunikation mit den Benutzern

Das Verzeichnis `/usr/news` und das Kommando `news` erlauben es, die Benutzer über aktuelle Ereignisse zu informieren. Tagesnachrichten können in die Datei `/etc/motd` (message of the day) eingegeben werden. Die Dateien `/etc/motd` und `/usr/news/*` werden nach der Anmeldung angezeigt.

Um alle Benutzer zu erreichen, auch die schon angemeldeten, wird das Kommando `wall` (`write all`) benutzt. `Wall` sollte vom Superuser nur in wirklich dringenden Fällen benutzt werden. Das Verzeichnis `/usr/news` sollte in regelmäßigen Abständen aufgeräumt werden. Erfahrungsgemäß erreicht eine Datei in `/usr/news` 50 % der Benutzer innerhalb eines Tages und 80 % der Benutzer innerhalb einer Woche.

Die Datei `motd` sollte täglich gelöscht und mit der neuen Tagesnachricht versehen werden.

4.5.2 Überwachung der Plattenverwendung

Es ist ratsam, die Plattenbelegung und den Füllgrad regelmäßig zu kontrollieren. Beim Tagesstart geben die Zähler Informationen über den derzeitigen Zustand. Mit den Kommandos `du` und `df` kann der gesamte Speicherplatz und die Größe der einzelnen Dateien und Verzeichnisse abgefragt werden. Diese Kommandos sollten mehrmals täglich ausgeführt werden und das Ergebnis in einer Datei für spätere Vergleiche festgehalten werden. Auf diese Art kann leicht ermittelt werden, bei welchem Benutzer bzw. in welchem Dateisystem die Plattenverwendung schnell zunimmt.

Systemverwaltung TARGON /31

Mit dem Kommando `find` können Sie besonders große oder lange nicht benutzte Dateien lokalisieren.

Beispiel:

```
find / -mtime +90 -atime +90 -print > verwaltung
```

Die Aufzeichnungen in der Datei `verwaltung` enthalten alle die Dateien, bei denen in den letzten 90 Tagen weder eine Änderung noch ein Zugriff erfolgte.

Als Administrator haben Sie auch die Aufgabe, die Ausnutzung des Platzes innerhalb der Dateisysteme zu überwachen. Um eine möglichst ausgewogene Auslastung zu erreichen, kann es nötig sein, Benutzer und ihre Dateien in andere Dateisysteme zu verlagern. Aus diesem Grund sollten Sie die Benutzer Ihres Systems dazu anhalten, ihre Programmierung auf eventuelle Dateisystemwechsel abzustellen. Z. B. ist es sinnvoll, die Shell Variable `HOME` zu verwenden, um Abhängigkeiten von Pfadnamen zu vermeiden.

Zur Verschiebung von ganzen Verzeichnisbäumen können Sie die Kommandos `find` und `cpio` benutzen. Das nachfolgende Beispiel zeigt, wie Sie die Benutzer `maria` und `doris` aus dem Dateisystem `doku` in das Dateisystem `doku1` schieben können:

```
cd /doku  
find . -user maria doris -print | cpio -pdm /doku1  
#Vergewissern Sie sich, ob das Kopieren geklappt hat  
#Ändern Sie die Login-Verzeichnisse von doris und maria  
#in der Datei /etc/passwd  
#Unterrichten Sie die betroffenen Benutzer mit der Aufforderung,  
#die nötigen Änderungen in ihren .profile-Dateien vorzunehmen  
rm -rf /doku/doris /doku/maria
```

Systemverwaltung TARGON /31

4.5.3 Verkleinern von Verzeichnissen

Verzeichnisse, deren Größe 4 KByte (= 256 Einträge) übersteigt, sind – aufgrund des indirekten Zugriffs – sehr uneffizient. Ein großes /usr/mail- oder /usr/spool/uucp-Verzeichnis kann also das System beträchtlich verlangsamen. Das folgende Kommando hilft Ihnen, solche übergroßen Verzeichnisse zu finden.

find / -type d -size +2 -print

Das Löschen von Dateien in Verzeichnissen bewirkt nicht, daß die Verzeichnisse kleiner werden, da die leeren Einträge zur Wiederbenutzung zur Verfügung bleiben. Die folgenden Kommandos zeigen am Beispiel von /usr/mail, wie man ein Verzeichnis verkleinern kann:

```
mv /usr/mail /usr/omail
mkdir /usr/mail
chmod 777 /usr/mail
cd /usr/omail
find . -print | cpio -plm ../mail
cd ..
rm -rf omail
```

Systemverwaltung TARGON /31

4.5.4 Reinigen der administrativen Log-Dateien

Die meisten der folgenden Dateien werden beim Booten durch Einträge in `/etc/rc` automatisch angelegt. Die Dateien werden sehr schnell sehr groß und müssen deshalb überwacht werden. Die nicht mehr benötigten Informationen sollten Sie regelmäßig löschen.

■ **Verwaltungsdateien:**

- `/usr/adm/wtmp`; wird benutzt für Login-Informationen; wächst extrem schnell bei Schwierigkeiten mit den Terminalleitungen. Diese Datei wird vom Accounting genutzt.
- `/usr/adm/pacct`; wird benutzt für Prozeß-Accounting; wächst schnell; wird automatisch überwacht durch `ckpacct` von `cron`.
- `/usr/adm/cronlog`; wird benutzt als Statuslogbuch von Kommandos, die durch `cron` ausgeführt werden. Diese Datei muß auf Fehlermeldungen von Programmen, die in `/usr/lib/crontab` stehen und ausgeführt werden, überprüft werden.
- `/usr/adm/errfile`; wird benutzt für Hardware-Fehlerprotokolle. Die Informationen sollten periodisch gelesen werden (siehe Administrator-Kommando `errpt`).

■ **Andere Dateien:**

- `/usr/spool`; ist das Spooling-Verzeichnis für Drucker; `uucp` usw. und dessen Unterverzeichnisse. Sie sollten wie oben beschrieben verkleinert werden.

Systemverwaltung TARGON /31

4.6 Zeitplan des Systemverwalters

Die Aufgaben des Systemverwalters müssen regelmäßig und sorgfältig ausgeführt werden, damit Systemausfälle weitestgehend vermieden werden.

Hier finden Sie beschrieben, welche administrativen Funktionen mit Hilfe des Programms cron ausgeführt werden. Außerdem wird ein Vorschlag unterbreitet, welche zusätzlichen Arbeiten vom Systemverwalter vorgenommen werden sollten.

Das Programm cron ist eine sehr nützliche Hilfe bei den administrativen Aufgaben. Es läuft als Hintergrundprozeß und liest einmal pro Minute die Datei /usr/lib/crontab, welche die Steuerparameter enthält, um zu sehen, ob augenblicklich irgendwelche Kommandos ausgeführt werden sollen. Damit können Aufgaben wie Datensicherung, Benutzerstatistik usw. regelmäßig erledigt werden, ohne daß sich der Systemverwalter ständig darum kümmert. Es wird z. B. für folgende Funktionen benutzt:

- Starten von Programmen außerhalb der normalen Arbeitszeit, um die Systemlast zu reduzieren.
 - Accounting
 - Verwaltung von Dateisystemen
 - Langlaufende Shell-Prozeduren

Nachfolgend wird ein Zeitplan für den Ablauf der manuellen Verwaltungsaufgaben gegeben:

- Täglich anfallende Arbeiten:
 - Überprüfung des freien Speicherplatzes
 - Überprüfung der Plattenauslastung
 - Ausführung der inkrementellen Sicherung

Systemverwaltung TARGON /31

- ≡ Wöchentlich anfallende Arbeiten:
 - Durchführen einer gesamten Sicherung
 - Überprüfung der Sicherung/Sicherheit
 - ≡ Regelmäßig anfallende Arbeiten (weniger oft als wöchentlich):
 - Verkleinern von Verzeichnissen
 - Verkleinern von wachsenden Dateien
 - Reorganisieren von Dateisystemen
 - Kommunikation mit den Benutzern über Probleme, z. B. ein bevorstehendes Herunterfahren, um den belegten Plattenplatz zu reorganisieren usw.
 - Zuweisung von Betriebsmitteln.
-

Das Spool-System

5 Das Spool-System

Dieses Kapitel beinhaltet einen Überblick über das Spool-System sowie die Ausführung geschützter Funktionen.

Das Spool-System ist ein Kommandosystem, das verschiedene Spooler-Funktionen unter dem TARGON /31-Betriebssystem ausführt.

Das Spool-System erlaubt den System-Administratoren die Drucker in logische Klassen zu gruppieren und somit eine optimale Ausnutzung der Geräte zu erzielen.

Um ein lauffähiges Spool-System zu installieren, müssen bestimmte Voraussetzungen erfüllt werden.

5.1 Konfiguration

Der Spooler unterscheidet zwischen Druckern und Gerätedateien. Eine Gerätedatei ist entweder ein physikalisches Peripheriegerät oder eine Datei und wird durch einen kompletten Pfadnamen repräsentiert. Zu verschiedenen Zeiten kann ein Drucker mit verschiedenen Namen assoziiert werden. Als Klasse bezeichnet man eine Anzahl von Druckern, die unter einem Namen definiert werden. Jede Klasse muß mindestens einen Drucker enthalten. Jeder Drucker kann Mitglied in mehreren Klassen sein. Er muß jedoch nicht unbedingt einer Klasse zugeordnet werden. Ein Druckauftrag kann an einen Drucker oder eine Klasse übergeben werden. Ein Drucker oder eine Klasse von Druckern können als Standardausgabegeräte definiert werden. Das lp-Kommando leitet diesem Druckgerät alle Ausgaben zu, die der Benutzer nicht ausdrücklich auf ein anderes Ausgabegerät umgeleitet hat. Eine Ausgabe, die auf einen bestimmten Drucker geleitet wird, wird nur von diesem gedruckt (z. B. ein Geschäftsbrief auf dem Schönschriftdrucker). Wird eine Klasse von Druckern angesprochen, so läuft der Druck auf dem ersten freien Drucker der Klasse, den das System findet.

Das Spool-System

Jeder Aufruf von `lp` erzeugt eine Ausgabeanforderung, die die zu druckenden Dateien sowie die Optionen der `lp`-Kommandozeile enthält. Für jeden Drucker muß ein Schnittstellenprogramm zur Formatierung der Druckwünsche vorhanden sein. Der Scheduler regelt für alle Druckaufträge die Verteilung auf die einzelnen Ausgabegeräte.

5.2 Kommandos

Im Spool-System stehen eine Reihe von Kommandos für Benutzer und Administratoren zur Verfügung.

5.2.1 Benutzerkommandos

Das `lp`-Kommando wird zur Erteilung von Druckaufträgen gebraucht. Es erzeugt die Ausgabeanforderungen und gibt als Rückmeldung eine Auftrags-ID in der Form:

Druckausgabegerät/Jobnummer

Das Kommando **cancel** löscht Druckaufträge. Sie übergeben die Auftrags-ID so, wie `lp` sie zurückgeliefert hat, um den Druckauftrag zu löschen. Es kann auch ein Druckername angegeben werden, in diesem Fall wird dessen aktueller Druckauftrag gelöscht.

Das Kommando **disable** hält den Scheduler davon ab, die Druckaufträge auf dem bezeichneten Drucker auszuführen.

Das Kommando **enable** ermöglicht dem Scheduler, den genannten Drucker wieder zu bedienen.

Eine ausführliche Beschreibung dieser Kommandos finden Sie in der Systemliteratur „Benutzerkommandos“.

Das Spool-System

5.2.2 Administrator-Kommandos

Jedes Spool-System braucht einen Systemverwalter für die Bedienung der unten aufgeführten Funktionen. Das ist in den meisten Fällen der Superuser, kann aber auch ein Benutzer sein, der sich unter dem Login-Namen „lp“ anmeldet. Die folgenden Kommandos werden im Verlauf dieser Dokumentation noch ausführlicher beschrieben (siehe Anhang „Administrator-Kommandos“). Normalerweise gehören alle Spooler-Dateien und -Kommandos dem Benutzer lp, nur lpadmin und lpsched gehören dem Superuser.

lpadmin	Konfiguration des Spool-Systems.
lpsched	Startet den Scheduler. Dieser übergibt die Druckaufträge an die Schnittstellenprogramme.
lpshut	Stoppt den Scheduler und alle Druckaktivitäten. Dieses Kommando wird etwa bei Konfigurationsänderungen gesetzt. Andere Spooler-Kommandos können trotzdem benutzt werden.
accept	Erlaubt lp Druckaufträge für einzelne Drucker bzw. Druckerklassen anzunehmen.
reject	Hält lp davon ab, Druckaufträge für einzelne Drucker bzw. Druckerklassen anzunehmen.
lpmove	Verschiebt komplette Druckaufträge zu anderen Klassen oder Druckern. Dieses Kommando kann nicht benutzt werden, wenn der Scheduler läuft.



Das Spool-System

5.3 Die Installation des Spool-Systems

Alle Mitglieder der Systemfamilie TARGON werden mit installiertem Spool-System ausgeliefert. Dieses Kapitel ist daher lediglich als ergänzende Information zu verstehen.

Als Administrator sollten Sie sicherstellen, daß in den Dateien `/etc/rc` und `/etc/shutdown` die folgenden Einträge enthalten sind:

1. `/etc/rc`

```
rm -f /usr/spool/lp/SCHEDLOCK
/usr/lib/lpsched
echo "Scheduler gestartet"
```

Durch diese Befehlsfolge wird der Scheduler bei jedem Hochfahren Ihres Systems gestartet.

2. `/etc/shutdown`

```
/usr/lib/lpshut
```

Dieses Kommando stoppt den Scheduler und beendet alle Druckaktivitäten.

5.3.1 Vorsichtsmaßnahmen

1. Einige Spooler-Kommandos rufen andere Spooler-Kommandos auf. Aus diesem Grund ziehen Verschiebungen oder Umbenennungen nach der Installation des Spoolers möglicherweise die fehlerhafte Ausführung einiger Kommandos nach sich.
2. Die Dateien im Spooler-Verzeichnis sollten nur mit Spooler-Kommandos verändert werden.
3. Alle Spooler-Kommandos verlangen eine Set-User-ID-Erlaubnis. Ist diese nicht erteilt, werden die Kommandos nicht ausgeführt.

Das Spool-System

5.4 Die Konfiguration des Spoolers mit lpadmin

Änderungen in der Spooler-Konfiguration sollten Sie mit dem lpadmin-Kommando vornehmen. Lpadmin ändert die Konfiguration nicht, wenn der Scheduler läuft, außer in ausdrücklich aufgeführten Sonderfällen.

5.4.1 Druckerkonfiguration

Zur Konfiguration von Druckern benötigt lpadmin verschiedene Informationen, die teils vorgeschrieben, teils optional sind.

5.4.1.1 Nicht-optionale Parameter

Die folgenden Informationen müssen zur Einbindung eines neuen Druckers an lpadmin übergeben werden:

1. Der Druckername (*-pDrucker*) ist willkürlich wählbar unter Beachtung folgender Regeln:
 - Er darf nicht länger als 14 Zeichen sein.
 - Er darf nur aus alphanumerischen Zeichen und dem Unterstrich () bestehen.
 - Es darf nicht der Name eines bereits bestehenden Druckers oder einer Klasse sein.
2. Die Gerätedatei, die mit dem Drucker verbunden ist (*-vGerätedatei*). Die Gerätedatei wird mit ihrem vollen Pfadnamen angegeben. Sie kann ein Drucker, ein Terminal oder eine Datei sein, die von Ip beschrieben werden kann.

Das Spool-System

3. Das Drucker-Schnittstellenprogramm. Es kann auf drei Arten spezifiziert werden:
 - Es kann aus einer Liste von Modellschnittstellenprogrammen ausgesucht werden, die das Spool-System zur Verfügung stellt (*-mModell*).
 - Es kann dasselbe Schnittstellenprogramm sein, das ein bereits vorhandener Drucker benutzt (*-eDrucker*).
 - Es kann ein vom Administrator geschriebenes Programm genutzt werden (*-iProgrammname*).

5.4.1.2 Optionale Parameter

Folgende Informationen können Sie optional zur Einbindung eines neuen Druckers angeben:

1. Ist das Gerät der Pfadname eines Terminals, sollten Sie *-l* in der Kommandozeile eingeben. Das zeigt dem Scheduler an, daß er diesem Gerät automatisch das Kommando *disable* übergeben muß, sobald der Scheduler gestartet wird. Dieser Umstand wird von *lpstat* – bei Ausgabe der Druckerstat – gemeldet.

```
$ lpstat -pa  
printer a (login terminal) disabled Oct 31 11:15-  
disabled by scheduler : login terminal
```

Wird die Eingabe von *-l* unterlassen, besteht die Möglichkeit, daß ein Benutzer sich anmeldet und an seinem Terminal werden vom Spooler dorthingeleitete Arbeiten ausgeführt.

2. Der neue Drucker kann einer bestehenden oder neuen Klasse von Druckern zugeordnet werden (*-cKlasse*). Neue Klassennamen sind den gleichen Regeln unterworfen wie neue Druckernamen.

Das Spool-System

5.4.1.3 Beispiele zur Druckerkonfigurierung

Auf die folgenden Beispiele beziehen sich auch Beispiele in späteren Abschnitten.

1. Legen Sie einen Drucker mit dem Namen dr1 an, dessen Pfadname /dev/printer und dessen Schnittstellenprogramm das Modell Ipsas ist:

```
$ /usr/lib/lpadmin -pdr1 -v /dev/printer -mlpsas
```

2. Konfigurieren Sie einen Drucker mit dem Namen dr2, dessen Pfadname /dev/ttyi22 und dessen Schnittstellenprogramm eine Variation des prx-Programms ist. Die Gerätedatei /dev/ttyi22 ist gleichzeitig ein Terminal:

```
$ cp /usr/spool/lp/model/prx xxx
< xxx editieren >
$ /usr/lib/lpadmin -pdr2 -v/dev/ttyi22 -ixxx -l
```

3. Legen Sie einen neuen Drucker mit dem Namen dr3 und dem Pfadnamen /dev/ttyi23 an. Der Drucker dr3 gehört zu einer neuen Klasse mit dem Namen cl1 und wird dasselbe Schnittstellenprogramm benutzen wie der Drucker dr2.

```
$ /usr/lib/lpadmin -pdr3 -v/dev/ttyi23 -edr2 -cc1
```



Das Spool-System

5.4.2 Änderung von Druckern oder Druckerklassen

Veränderungen von bestehenden Druckern oder Druckerklassen müssen immer mit Bezug auf einen Druckernamen (*-pDrucker*) gemacht werden. Sie können jeweils eine oder mehrere der folgenden Änderungen vornehmen:

- Änderung der Gerätezuordnung für den Drucker durch *-vPfadname*. Dies ist die einzige Änderung, die zur Laufzeit durchgeführt werden kann. Das erleichtert das Ändern der Gerätezuordnung von Terminals.
- Wechseln des Schnittstellenprogramms (*-mModell*, *-eDrucker*, *-iSchnittstelle*).
- Änderung der Druckerspezifizierung (*-h* oder *-l*).
- Zuordnung der Drucker zu einer bestehenden oder neuen Klasse (*-cKlasse*).
- Entfernung des angegebenen Druckers aus dem Spool-System (*-x Drucker*).
- Entfernung von Druckern aus einer bestehenden Klasse (*-rKlasse*). Wird der letzte Drucker aus einer Klasse entfernt, so wird die Klasse gelöscht.
- Kein Drucker darf aus einer Klasse entfernt werden, solange für ihn noch Aufträge vorliegen. In diesem Fall sollten Sie die Aufträge mit dem Kommando *cancel* löschen oder mit *lpmove* umleiten.

Das Spool-System

5.4.3 Beispiele zur Änderung von Druckern

Diese Beispiele basieren auf der Konfiguration, die im vorhergehenden Abschnitt geschaffen wurde.

1. Fügen Sie den Drucker dr2 der Klasse cl1 zu:

```
$ /usr/lib/lpadmin -pdr2 -ccl1
```

2. Ändern Sie das Schnittstellenprogramm des Druckers dr2 in das Modell prx, ändern Sie seine Gerätezuordnung auf /dev/ttyi24 und fügen Sie ihn in eine neue Klasse mit dem Namen cl2 ein:

```
$ /usr/lib/lpadmin -pdr2 -mprx -v/dev/ttyi24 -ccl2
```

Beachten Sie, daß die Drucker dr2 und dr3 jetzt unterschiedliche Schnittstellenprogramme benutzen, obwohl dr3 ursprünglich mit demselben Schnittstellenprogramm wie dr2 angelegt wurde. Der Drucker dr2 ist jetzt Mitglied in zwei Klassen.

3. Spezifizieren Sie dr2 als Drucker:

```
$ /usr/lib/lpadmin -pdr2
```

4. Fügen Sie den Drucker dr1 der Klasse cl2 zu:

```
$ /usr/lib/lpadmin -pdr1 -ccl2
```

Die Drucker dr2 und dr1 sind nun in dieser Reihenfolge Mitglieder der Klasse cl2. Sind beide Drucker bereit, werden Druckaufträge an diese Klasse vom Drucker dr2 ausgeführt. Ist einer oder sind beide Drucker belegt, wird der Auftrag von dem Drucker ausgeführt, der als erster wieder frei ist.

5. Entfernen Sie die Drucker dr2 und dr3 aus der Klasse cl1:

```
$ /usr/lib/lpadmin -pdr2 -rccl1
```

```
$ /usr/lib/lpadmin -pdr3 -rccl1
```

Dr3 war der letzte Drucker der Klasse, mit seiner Entfernung wird die Klasse cl1 gelöscht.

6. Ordnen Sie den Drucker dr3 einer neuen Klasse mit dem Namen cl3 zu:

```
$ /usr/lib/lpadmin -pdr3 -ccl3
```



Das Spool-System

5.4.4 Einrichten des Standard-Systemdruckers

Der Standard-Systemdrucker kann auch geändert werden, wenn der Scheduler aktiv ist.

Beispiele

1. Einrichten der Klasse cl1 als Standardausgabegerät:

```
$ /usr/lib/lpadmin -dcl1
```

2. Befehl, der im System kein Standardausgabegerät definiert:

```
$ /usr/lib/lpadmin -d
```

5.4.5 Löschen von Druckern oder Klassen

Klassen und Drucker können nur gelöscht werden, wenn keine Druckaufträge an sie adressiert sind. Schwebende Druckaufträge müssen entweder mit dem Kommando cancel gelöscht oder mit dem Kommando lpmove umgeleitet werden, bevor eine Löschung des Druckers oder der Klasse erfolgen kann. War der gelöschte Drucker oder die gelöschte Klasse das Standardausgabegerät, hat das System kein Standardausgabegerät mehr, bevor ein neues definiert wird. Löschen Sie den letzten Drucker einer Klasse, so wird auch die Klasse gelöscht. Das Löschen einer Klasse beinhaltet jedoch nicht das Löschen der darin enthaltenen Drucker.

Beispiele:

1. Setzen Sie den Drucker dr1 als Standarddruckausgabegerät:

```
$ /usr/lib/lpadmin -ddr1
```

Löschen Sie den Drucker dr1:

```
$ /usr/lib/lpadmin -xdr1
```

Das System hat jetzt kein Standarddruckausgabegerät mehr.

Das Spool-System

2. Löschen Sie den Drucker dr2:

```
$ /usr/lib/lpadmin -xdr2
```

Da der Drucker dr2 das einzige Mitglied der Klasse cl2 war, wird diese automatisch mitgelöscht.

3. Löschen Sie die Klasse cl3:

```
$ /usr/lib/lpadmin -xcl3
```

Die Klasse cl3 wurde gelöscht, aber der Drucker dr3 ist weiterhin konfiguriert.

5.5

Der Scheduler

In diesem Kapitel finden Sie nähere Informationen über den Scheduler sowie Beispiele zum Starten und Stoppen des Schedulers.

Der Scheduler leitet alle mit lp angelegten Druckaufträge über die Schnittstellenprogramme zur Verarbeitung an die Druckausgabegeräte. Immer wenn der Scheduler einen Druckauftrag zu einem Schnittstellenprogramm leitet, erfolgt ein Eintrag in die Log-Datei /usr/spool/lp/log. Dieser Eintrag enthält den Benutzernamen des Auftraggebers, die Druckauftrags-ID, den Namen des gewünschten Druckers (oder der Klasse) und das Datum und die Zeit, wann der Auftrag erteilt wurde. Wird die Ausführung des Auftrags unterbrochen und neu gestartet, ist es möglich, daß sich mehrere Einträge in der Logdatei auf einen Auftrag beziehen. Der Scheduler schreibt auch Fehlermeldungen in die Logdatei. Starten Sie den Scheduler, benennt er /usr/lib/lp/log in /usr/lib/lp/oldlog um und startet eine neue Logdatei.

Solange der Scheduler nicht läuft, wird kein Druckauftrag ausgeführt. Benutzen Sie das Kommando lpstat -r, um den Status des Schedulers festzustellen.



Das Spool-System

5.5.1 Starten des Schedulers durch lpsched

Lpsched wird normalerweise im Programm /etc/rc gestartet und läuft, bis das System heruntergefahren wird. Der Scheduler operiert im /usr/spool/lp-Verzeichnis. Wenn er gestartet wird, überprüft er als erstes, ob eine Datei mit dem Namen SCHEDLOCK existiert. Ist dies der Fall, terminiert er sofort. Andernfalls legt er die Datei SCHEDLOCK an. Diese Überprüfung verhindert, daß mehr als ein Scheduler zur gleichen Zeit läuft.

Um den Scheduler nach einem Stop neu zu starten, benutzen Sie das Kommando

```
/usr/lib/lpsched
```

Kurz nachdem das Kommando abgesetzt ist, sollte lpstat anzeigen, daß der Scheduler läuft. Falls nicht, wurde der vorhergehende Scheduler gestoppt, ohne daß die Datei SCHEDLOCK gelöscht wurde. In diesem Fall führen Sie folgendes aus:

```
rm -f /usr/spool/lp/SCHEDLOCK  
/usr/lib/lpsched
```

Jetzt sollte der Scheduler laufen.

5.5.2 Stoppen des Schedulers durch lpshut

Manchmal ist es notwendig, den Scheduler anzuhalten, um den Spooler zu rekonfigurieren oder Software wieder aufzubauen. Das Kommando

```
/usr/lib/lpshut
```

stoppt den Scheduler und beendet alle Druckaktivitäten. Alle unterbrochenen Druckaufträge werden noch einmal vollständig ausgegeben, sobald der Scheduler wieder gestartet ist.

Das Spool-System

5.6 Schnittstellenprogramme für Drucker

Jeder Drucker im Spool-System muß ein Schnittstellenprogramm haben, das den Ausdruck in eine gewünschte Form und Bahn lenkt. Schnittstellenprogramme können Shell-Prozeduren, C-Programme oder andere ausführbare Programme sein. Die Spooler-Modellschnittstellen sind ausnahmslos Shell-Prozeduren und stehen in dem Verzeichnis /usr/spool/lp/model. Zu dem Zeitpunkt, an dem lpsched einen Druckauftrag an einen Drucker lenkt, wird das Schnittstellenprogramm für diesen Drucker im Verzeichnis /usr/spool/lp wie folgt aufgerufen:

interface/Drucker ID Benutzername Titel Kopien Optionen Dateiname

Die entsprechenden Felder enthalten folgende Informationen:

Auftrags-ID	wie von lp zurückgeliefert
Benutzername	Login-Name des Auftraggebers
Titel	Durch den Benutzer spezifizierter Titel des Ausdrucks
Kopien	Anzahl der Kopien
Optionen	Optionen für die Druckaufbereitung
Dateiname	voller Pfadname der zu druckenden Datei



Das Spool-System

5.6.1 Generierung von Kommandozeilen

Die folgenden Beispiele sind Druckaufträge des Benutzers robert. Standarddruckausgabe ist der Drucker xyz. Jedes Beispiel zeigt zuerst die lp-Kommandozeile, die der Benutzer robert eingibt und dann die entsprechende Kommandozeile, die im Schnittstellenprogramm generiert wurde:

1. lp /etc/group
interface/xyz xyz-52 robert " " 1 " " /etc/group
2. pr /etc/group | lp -t" dokumentation" -n5
interface/xyz xyz-53 robert dokumentation 5 " "
/usr/spool/lp/request/xyz/d0-53
3. lp /etc/group -oa -ob
interface/xyz xyz-54 robert " " 1 "a b" /etc/group

5.6.2 Ausgabe mit Hilfe von Schnittstellenprogrammen

Wird ein Schnittstellenprogramm aufgerufen, ist seine Standardeingabe die Datei /dev/null und Standardausgabe und Standardfehlerausgabe werden auf das gewünschte Druckausgabegerät gelenkt. Ist eine Gerätedatei eine reguläre Datei, werden alle Ausgaben an das Dateiende angehängt.

Sind in der Kommandozeile ein Druckausgabegerät und verschiedene Optionen angegeben, formatiert das Schnittstellenprogramm den Ausdruck in der gewünschten Weise. Schnittstellenprogramme müssen sicherstellen, daß die speziellen Terminal-Modi (Terminal-Charakteristika wie die Baud-Rate, Ausgabe-Optionen etc.) entsprechend auf dem Druckausgabegerät eingestellt sind. Dies kann folgendermaßen, für ein Gerät, daß nur zum Lesen eröffnet ist, eingestellt werden:

```
stty <Modus> ...&1
```

Das heißt, daß die Standardeingabe für das stty-Kommando das Druckausgabegerät ist.

Das Spool-System

5.6.3 Rückgabecodes der Schnittstellenprogramme

Ist ein Druckauftrag beendet, leitet das Schnittstellenprogramm einen Code über den Erfolg des Druckauftrags zurück. Diese Codes werden von Ipsched wie folgt interpretiert:

Code Bedeutung für Ipsched

0 Der Druckauftrag wurde erfolgreich durchgeführt.

1-127 Während des Ausdrucks ist ein Fehler aufgetreten, der sich jedoch nicht auf die folgenden Druckaufträge auswirkt. Ipsched benachrichtigt den Benutzer per mail, daß ein Fehler in dem Druckauftrag aufgetreten ist.

>127 Diese Codes sind für den internen Gebrauch von Ipsched reserviert. Schnittstellenprogramme dürfen keinen Code in diesem Bereich zurückliefern.

Tritt ein Fehler auf, der auch folgende Druckaufträge beeinträchtigt, sollte das Schnittstellenprogramm den Drucker sperren (disable), damit die Druckaufträge nicht verlorengehen. Wenn ein arbeitender Drucker den Status disable erhält, wird das Schnittstellenprogramm durch Signal 15 beendet.



Das Spool-System

5.7 Einsatz von Geräten und Terminals als Drucker

Im UNIX-Spool-System können sowohl verschiedene Ausgabegeräte als auch Terminals als Drucker konfiguriert werden.

5.7.1 Geräte

Als Beispiel für den Einsatz von Geräten als Drucker richten Sie die Terminalleitung 15 (VT 100) als Drucker xyz ein. Führen Sie als Superuser folgende Dinge aus:

1. Stellen Sie sicher, daß keine unerwünschten Ausgaben (von nicht-Spooler-Prozessen) auf die Leitung kommen, und daß der Spooler darauf lesen und schreiben kann:

```
$ chown lp /dev/ttyi15
$ chmod 600 /dev/ttyi15
```

2. Ändern Sie /etc/inittab so, daß ttyi15 nicht als Terminal behandelt wird. Mit anderen Worten, daß /etc/getty nicht versucht, Benutzer an diesem Terminal einzuloggen. Ändern Sie die Einträge für die Leitung 15 wie folgt:

```
1:3:respawn:/etc/getty ttyi15 9600 vt100
```

Geben Sie folgendes Kommando ein:

```
$ init q
```

Ist ein aktueller getty-Aufruf für Leitung 15 vorhanden, muß dieser mit dem kill-Kommando terminiert werden. Wenn jetzt das UNIX-System neu gestartet wird, wird ttyi15 mit den Standard-stty-Modi initialisiert. Damit obliegt es den Schnittstellenprogrammen, die richtige Baud-Rate und andere Terminal-Modi zu übergeben um ein korrektes Drucken zu gewährleisten.

3. Benutzen Sie das Schnittstellenprogramm prx für den Drucker xyz:

```
$ /usr/lib/lpadmin -pxyz -v/dev/ttyi15 -mprx
```

Das Spool-System

4. Wenn xyz angelegt ist, wird das Spool-System zunächst keine Druckaufträge für ihn annehmen. Dies können Sie durch das folgende Kommando ändern:

```
/usr/lib/accept xyz
```

5. Wünschen Sie die Ausführung der Druckaufträge, stellen Sie sicher, daß der mit der Leitung 15 assoziierte Drucker empfangsbereit ist. Ändern Sie den Status des Druckers auf enable:

```
enable xyz
```

Bei anstehenden Druckaufträgen für xyz werden diese jetzt gedruckt.

5.7.2 Terminals

Terminals können ebenfalls als Drucker behandelt werden. Um ein Terminal abc als Drucker einzubinden, geben Sie folgenden Befehl ein:

```
$ /usr/lib/lpadmin -pabc -v/dev/null -idumb -l
```

Der Drucker abc wird mit dem Schnittstellenprogramm dumb angelegt. Nach dem Anlegen hat abc zunächst den Status disable. Außerdem akzeptiert lp keine Aufträge für abc. Durch das folgende Kommando autorisieren Sie lp zur Annahme von Druckaufträgen für abc:

```
$ /usr/lib/accept abc
```

Jetzt kann lp Druckaufträge für abc annehmen. Es empfiehlt sich jedoch, folgende Schritte durchzuführen, bevor Sie abc den Status enabled zuweisen:

1. Loggen Sie das Terminal ein.
2. Verbinden Sie das Terminal (angenommener Pfadname /dev/tty02) mit dem Drucker abc:

```
$ /usr/lib/lpadmin -pabc -v/dev/tty02
```

Achten Sie darauf, daß Sie lpadmin nur als Spooler-Administrator aufrufen können!



Das Spool-System

3. Jetzt geben Sie das Ausgabegerät frei:

```
enable abc
```

Stehen Aufträge für abc an, werden sie jetzt gedruckt.

4. Wollen Sie abc wieder als Terminal nutzen, setzen Sie den Status wieder auf disabled:

```
disable abc
```

Beim Starten des Systems oder beim Start des Schedulers erhält abc automatisch den Status disabled.

Anhang 1: Administrator-Kommandos

A1 Administrator-Kommandos

accept, reject – Annahme von Druckaufträgen freigeben/sperren

SYNTAX:

```
/usr/lib/accept Ziele
/usr/lib/reject [-r[Grund]] Ziele
```

BESCHREIBUNG:

Das **accept**-Kommando erlaubt dem Kommando lp für die angegebenen Ziele Druckaufträge anzunehmen. *Ziel* kann ein Drucker oder eine Klasse von Druckern sein. Mit Hilfe des Kommandos lpstat können Sie den Status der Ziele feststellen.

Durch **reject** untersagen Sie dem lp-Kommando Druckaufträge für die angegebenen Ziele anzunehmen. *Ziel* kann ein Drucker oder eine Klasse von Druckern sein. Mit lpstat können Sie den Status der Ziele feststellen.

Option:

-r[*Grund*] Angabe des Grundes für die Sperrung. Dieser Grund gilt für alle Drucker bis zur nächsten Option -r. Der Grund wird zum einen von lp ausgegeben, wenn ein Benutzer Aufträge für die gesperrten Ziele absetzt und zum ändern von lpstat. Geben Sie die Option -r nicht oder ohne *Grund* an, wird ein Standardtext verwendet.

DATEIEN:

```
/usr/spool/lp/*
```

SIEHE AUCH:

Administrator-Kommandos: lpadmin, lpsched
Benutzerkommandos: enable, lp, lpstat

Anhang 1: Administrator-Kommandos

acctdisk, acctdusg, accton, acctwtmp – Accounting-Überblick und Beschreibung von Accounting-Kommandos

SYNTAX:

```
/usr/lib/acct/acctdisk  
/usr/lib/acct/acctdusg [-u Datei] [-p Datei]  
/usr/lib/acct/accton [Datei]  
/usr/lib/acct/acctwtmp "Grund"
```

BESCHREIBUNG:

Die Accounting-Software besteht aus einer Reihe von Tools (C-Programmen und Shell-Prozeduren), mit denen der Anwender Accounting-Systeme aufbauen kann. Unter acctsh sind die Shell-Prozeduren beschrieben, die auf die C-Programme aufgesetzt sind.

Die Abrechnung der Anschlußzeit erfolgt über verschiedene Programme, die ihre Werte in /etc/utmp schreiben. Die unter acctcon beschriebenen Programme konvertieren den Inhalt dieser Datei in Sessions- und Gebührensätze, die anschließend von acctmrg summiert werden.

Prozeß-Accounting wird vom UNIX-Systemkern abgewickelt. Bei Beendigung eines Prozesses wird ein Datensatz pro Prozeß in eine Datei (in der Regel /usr/adm/pacct) geschrieben. Die Programme in acctprc fassen diese Daten für Abrechnungszwecke zusammen; mit acctcms erhält man einen Überblick über die Häufigkeit der Benutzung von Kommandos. Die aktuellen Prozeßdaten können mit Hilfe von acctcom untersucht werden.

Prozeß- und Anschlußzeit-Accounting (und alle Accounting-Sätze im Format der acct-Datei) können mit Hilfe von acctmrg zusammengeführt und zu Summen-Abrechnungssätzen zusammengefaßt werden (siehe tacct-Format in acct-Format). Mit prtacct (siehe acctsh) werden alle oder ausgewählte Abrechnungssätze formatiert.

Anhang 1: Administrator-Kommandos

Acctdisk liest Zeilen, die eine Benutzer-ID, einen Login-Namen und eine Anzahl Plattenblöcke enthalten und konvertiert sie in Summen-Abrechnungssätze, die mit anderen Abrechnungssätzen zusammengeführt werden können.

Acctdusg liest seine Standardeingabe (normalerweise von find / -print) und errechnet den Verbrauch an Plattenspeicherplatz (einschließlich indirekter Blöcke) durch login. Ist -u angegeben, werden Datensätze mit den Dateinamen, für die acctdusg niemanden belastet, in der angegebenen Datei abgestellt. (Dies ist eine Möglichkeit, Benutzer ausfindig zu machen, die die Plattenbenutzungsgebühren umgehen wollen.) Ist -p angegeben, ist die angegebene Datei der Name der Paßwortdatei. Diese Option ist nicht erforderlich, wenn die Paßwortdatei /etc/passwd ist. (Weitere Angaben siehe diskusg.)

Accton allein deaktiviert die Prozeßabrechnung. Geben Sie eine Datei an, muß dies der Name einer vorhandenen Datei sein, an die der Betriebssystemkern die Prozeßabrechnungssätze anfügt (siehe Systemaufruf acct und Dateiformat acct).

Acctwtmp schreibt einen utmp-Satz in die Standardausgabe. Der Datensatz enthält die aktuelle Zeit und eine Zeichenfolge, die den Grund angibt. Der Satztyp ACCOUNTING wird zugewiesen (siehe Dateiformat utmp). *Grund* muß eine Folge von maximal 11 Buchstaben, Ziffern, \$ oder Leerzeichen sein. Die folgenden Beispiele sind Vorschläge für die Anwendung in Prozeduren für Neustart bzw. Herunterfahren:

```
acctwtmp 'uname' >> /etc/wtmp
acctwtmp "file save" >> /etc/wtmp
```

DATEIEN:

/etc/passwd	Wird für die Konvertierung der Login-Namen in Benutzer-IDs verwendet.
/usr/lib/acct	Enthält alle Administrator-Abrechnungskommandos.
/usr/adm/pacct	Aktuelle Datei für Prozeß-Accounting.
/etc/wtmp	Protokolldatei für Anmelden/Abmelden.

Anhang 1: Administrator-Kommandos

SIEHE AUCH:

Administrator-Kommandos: acctcms, acctcon, acctmerg, acctprc,
acctsh, diskusg, fwtmp, runacct

Dateiformate: acct, utmp

Benutzerkommando: acctcom

Systemschnittstelle: acct

Anhang 1: Administrator-Kommandos

acctcms – Zusammenfassung von Prozeß-Abrechnungssätzen

SYNTAX:

/usr/lib/acct/acctcms [Optionen] Dateien

BESCHREIBUNG:

Acctcms liest eine oder mehrere Dateien normalerweise in der Form, wie sie im Dateiformat acct beschrieben ist. Das Kommando schreibt alle Sätze für Prozesse, die Kommandos mit identischen Namen ausgeführt haben, in die Datei, sortiert sie und gibt sie auf der Standardausgabe aus, wobei in der Regel ein internes Format für die Zusammenfassung verwendet wird.

Optionen:

- a Die Daten werden in ASCII statt im internen Zusammenfassungsformat ausgegeben. Bei den Ausgabedaten handelt es sich um den Kommandonamen, die Anzahl der Ausführungen, die Summe der kcore-Minuten, die Summe der CPU-Minuten, die Summe der realen Minuten, die mittlere Größe (in K), die mittleren CPU-Minuten pro Aufruf und die CPU-Zeit, die ein Prozeß während seiner Ausführung benötigt (hog factor), wie beim Benutzerkommando acctcom. Die Angaben werden normalerweise nach der Summe der genauen Speicherbelegungszeit in Minuten (kcore minutes) sortiert.
- c Die Sortierung erfolgt nach der Summe der CPU-Zeit und nicht nach der Summe der kcore-Minuten.
- j Alle Kommandos, die nur einmal aufgerufen wurden, werden unter ***other zusammengefaßt.
- n Die Sortierung erfolgt nach der Anzahl der Kommandoaufrufe.
- s Alle hiernach angegebenen Dateinamen haben bereits das interne Zusammenfassungsformat.

Anhang 1: Administrator-Kommandos

Nachfolgend finden Sie ein Beispiel für eine typische Kommandofolge, mit der Sie die tägliche Kommandoabrechnung durchführen und die laufende Summe aktualisieren können:

```
acctcms Datei ... >heute
cp summe summealt
acctcms -s heute summealt >summe
acctcms -a -s heute
```

SIEHE AUCH:

Administrator-Kommandos: acct, acctcon, acctmerg, acctprc, acctsh, fwtmp, runacct

Dateiformate: acct, utmp

Benutzerkommando: acctcom

Systemschnittstelle: acct

Anhang 1: Administrator-Kommandos

acctcon1, acctcon2 – Abrechnung von Anschlußzeiten

SYNTAX:

```
/usr/lib/acct/acctcon1 [Optionen]
/usr/lib/acct/acctcon2
```

BESCHREIBUNG:

Acctcon1 konvertiert eine Folge von Login/Logoff-Sätzen, die aus der Standardeingabe gelesen werden, in eine Folge von Sätzen mit einem Satz pro Login-Session. Die Eingabe muß normalerweise von /etc/wtmp umgeleitet werden. Die Ausgabedaten haben ASCII-Format und umfassen das Gerät, die Benutzer-ID, den Login-Namen, die Hauptanschlußzeit (in Sekunden), die Nicht-Hauptanschlußzeit (in Sekunden), die Startzeit der Session (numerisch) sowie das Startdatum und die Startzeit.

Optionen:

- p Nur die Eingabedaten ausgeben und dabei den Leitungsnamen, den Login-Namen und die Zeit (sowohl im numerischen als auch im Datum-/Zeitformat) angeben.
- t Acctcon1 führt eine Liste der Leitungen, auf denen Benutzer angemeldet sind. Erkennt acctcon1 das Ende seiner Eingabedaten, setzt es einen Session-Satz für jede Leitung, die noch als aktiv erkannt wird, ab. Acctcon1 geht normalerweise davon aus, daß seine Eingabe eine aktuelle Datei ist, und es setzt die aktuelle Zeit als Beendigungszeit für alle noch laufenden Sessions ein. Bei Angabe der Option -t setzt das Kommando stattdessen die letzte in seiner Eingabe aufgetretene Zeit ein und gewährleistet so sinnvolle und wiederholbare Angaben für nicht-aktuelle Dateien.

Anhang 1: Administrator-Kommandos

- l *Datei* In der angegebenen Datei wird ein Überblick über die Leitungsbelegung gespeichert. Die Datensätze enthalten den Leitungsnamen, die Belegungszeit in Minuten, den Prozentsatz der belegten Zeiten gegenüber der Gesamtzeit, die Anzahl der weiterbelasteten Sessions, die Anzahl der Anmeldungen und die Anzahl der Abmeldungen. Mit Hilfe dieser Datei kann die Leitungsbelegung überwacht, gestörte Leitungen erkannt und Besonderheiten der Software und Hardware ausfindig gemacht werden. Jedes „Hang-Up“, und jedes Terminieren von login oder der Login-Shell erzeugen einen Abmeldesatz, so daß häufig drei- bis viermal so viele Abmeldungen wie Sessions verzeichnet werden. (Siehe auch Administrator-Kommando init und Dateiformat utmp.)
- o *Datei* In die angegebene Datei wird ein Gesamtsatz für die Abrechnungsperiode geschrieben, der die Startzeit, die Beendigungszeit, die Anzahl der Neustarts und die Anzahl der Datumswechsel enthält.

Acctcon2 erwartet als Eingabe eine Folge von Anmeldesätzen und konvertiert sie in Summen-Abrechnungssätze (siehe Format tacct im Dateiformat acct).

BEISPIEL:

Die Kommandos werden typischerweise in der unten angegebenen Form verwendet. Die Datei ctmp wird nur für den Gebrauch des Administrator-Kommandos acctprc angelegt:

```
acctcon1 -t -l Leitungsbel -o Neustarts <wtmp | sort +1n +2  
>ctmp  
acctcon2 <ctmp | acctmerg >ctacct
```

DATEIEN:

/etc/wtmp

Anhang 1: Administrator-Kommandos

HINWEIS:

Die Auswertung der Leitungsbelegung wird durch Datumswechsel verfälscht. Zur Korrektur sollten Sie in diesem Fall wtmpix (s. Administrator-Kommando fwtmp) verwenden.

SIEHE AUCH:

Administrator-Kommandos: acct, acctcms, acctmerg, acctprc, acctsh, fwtmp, init, runacct

Dateiformat: acct, utmp

Benutzerkommandos: acctcom, init, login

Systemaufruf: acct

Anhang 1: Administrator-Kommandos

acctmerg – Abrechnungsdateien zusammenfassen oder addieren

SYNTAX:

```
/usr/lib/acct/acctmerg [Optionen] [Datei] ...
```

BESCHREIBUNG:

Acctmerg liest seine Standardeingabe und bis zu neun weitere Dateien im tacct-Format (siehe Dateiformat acct) oder in einer ASCII-Version davon. Es mischt diese Eingaben durch Addieren der Datensätze, deren Ordnungsbegriffe (normalerweise die Benutzer-ID und der Benutzername) identisch sind, und erwartet, daß die Eingaben nach diesen Ordnungsbegriffen sortiert sind.

Optionen:

- a Die Ausgabedaten werden in der ASCII-Version von tacct ausgegeben.
- i Die Eingabedateien sind in der ASCII-Version von tacct.
- p Die Eingabedaten werden ohne weitere Verarbeitung ausgegeben.
- t Erstellung eines einzigen Datensatzes, der alle Eingaben summiert.
- u Zusammenfassung nur nach Benutzer-ID statt nach Benutzer-ID und Benutzername.
- v Ausgabe im ausführlichen ASCII-Format mit genauerer Notation der Gleitkommazahlen.

Die folgende Kommandofolge hilft Ihnen bei der „Reparatur“ aller Dateien dieses Formats:

```
acctmerg -v <Datei1 > Datei2  
# Datei2 wie gewünscht überarbeiten ...  
acctmerg -i <Datei2 > Datei1
```

Anhang 1: Administrator-Kommandos

SIEHE AUCH:

Administrator-Kommandos: acct, acctcms, acctcon, acctprc, acctsh,
fwtmp, runacct

Dateiformate: acct, utmp

Benutzerkommando: acctcom

Systemschnittstelle: acct

Anhang 1: Administrator-Kommandos

acctprc1, acctprc2 – Prozeßabrechnung

SYNTAX:

```
/usr/lib/acct/acctprc1 [ctmp]
/usr/lib/acct/acctprc2
```

BESCHREIBUNG:

Acctprc1 liest Eingabedaten entsprechend dem Dateiformat acct, fügt die den Benutzer-IDs entsprechenden Login-Namen hinzu und gibt dann für jeden Prozeß eine ASCII-Zeile mit Benutzer-ID, Login-Name, primärer CPU-Zeit (Takte), nicht-primäre CPU-Zeit (Takte) und die mittlere Speichergröße (in Einheiten von 64 Bytes) aus. Ist ctmp angegeben, muß darin eine Liste der Anmeldesessionen in der in acctcon beschriebenen Form enthalten sein, die nach Benutzer-ID und Login-Name sortiert ist. Wird diese Datei nicht als Argument übergeben, holt sich acctprc1 die Login-Namen aus der Paßwortdatei. Aufgrund der Angaben in ctmp kann acctprc1 aber verschiedene Login-Namen, die dieselbe Benutzer-ID haben, unterscheiden.

Acctprc2 liest Sätze in der Form, wie sie von acctprc1 ausgegeben werden, faßt sie nach Benutzer-ID und Benutzernamen zusammen und gibt die sortierten Zusammenfassungen als Summen-Abrechnungssätze auf der Standardausgabe aus.

Diese Kommandos werden meistens in der folgenden Form verwendet:

```
acctprc1 ctmp </usr/adm/pacct | acctprc2 >ptacct
```

DATEIEN:

```
/etc/passwd
```

Anhang 1: Administrator-Kommandos

HINWEIS:

Obwohl es durchaus möglich ist, verschiedene Login-Namen unter der gleichen Benutzer-ID bei Kommandoausführungen zu unterscheiden, ist diese Unterscheidung, z. B. bei Kommandos, die von cron ausgeführt werden, schwierig. Eine genauere Konvertierung kann man erzielen, indem man Anmeldesessionen an der Konsole über das Programm `acctwtmp` in `acct` simuliert.

SIEHE AUCH:

Administrator-Kommandos: `acct`, `acctcms`, `acctcon`, `acctmrg`, `acctsh`,
`cron`, `fwtmp`, `runacct`
Dateiformate: `acct`, `utmp`
Benutzerkommando: `acctcom`
Systemschnittstelle: `acct`

Anhang 1: Administrator-Kommandos

chargefee, ckpacct, dodisk, lastlogin, monacct, nulladm, prctmp, prdaily, prtacct, runacct, shutacct, startup, turnacct – Shell-Prozeduren für Accounting

SYNTAX:

```
/usr/lib/acct/chargefee Login-Name Anzahl  
/usr/lib/acct/ckpacct [Blöcke]  
/usr/lib/acct/dodisk [-o] [Datei ...]  
/usr/lib/acct/lastlogin  
/usr/lib/acct/monacct [Monat]  
/usr/lib/acct/nulladm Datei  
/usr/lib/acct/prctmp  
/usr/lib/acct/prdaily [-l] [-c] [MMTT]  
/usr/lib/acct/prtacct Datei ["Überschrift"]  
/usr/lib/acct/runacct [MMTT] [MMTT Status]  
/usr/lib/acct/shutacct ["Grund"]  
/usr/lib/acct/startup  
/usr/lib/acct/turnacct on | off | switch
```

BESCHREIBUNG:

Durch Aufruf von **chargefee** wird der angegebene Login-Name mit der entsprechenden Anzahl von Einheiten belastet. Dazu wird ein Datensatz in `/usr/adm/fee` geschrieben, der während der Nacht mit anderen Abrechnungssätzen gemischt wird.

Anhang 1: Administrator-Kommandos

Ckpacct sollte von cron gestartet werden. Dieses Kommando prüft periodisch die Größe von /usr/adm/pacct. Übersteigt die Größe die angegebene Anzahl von Blöcken (Standard: 1000 Blöcke), wird turnacct mit dem Argument switch aufgerufen. Sinkt die Anzahl der freien Plattenblöcke im Dateisystem /usr unter 500, stoppt ckpacct die Erfassung von Prozeßabrechnungssätzen automatisch mit Hilfe des Arguments off von turnacct. Ist mindestens diese Anzahl Blöcke wieder vorhanden, wird die Abrechnung wieder aktiviert. Diese Funktion hängt davon ab, wie oft ckpacct angestoßen wird. Dies geschieht gewöhnlich über cron.

Dodisk sollte von cron aufgerufen werden. Es führt die Plattenabrechnungsfunktionen aus. Standardmäßig wird die Abrechnung für die Spezialdateien durchgeführt, die in /etc/checklist eingetragen sind. Geben Sie die Option -o an, wird eine langsamere Version der Plattenabrechnung für das Login-Verzeichnis durchgeführt. Die Angabe von *Dateien* spezifiziert ein oder mehrere Dateisysteme, in denen die Abrechnungsfunktionen durchgeführt werden.

Lastlogin wird von runacct aufgerufen. Diese Prozedur aktualisiert die Datei /usr/adm/acct/sum/loginlog. Sie enthält Zeitangaben über die jeweils letzte Anmeldung der einzelnen Benutzer.

Monacct sollte einmal monatlich bzw. einmal pro Abrechnungsperiode aufgerufen werden. Das Argument gibt den gewünschten Monat bzw. die Abrechnungsperiode an. Geben Sie kein Argument an, wird standardmäßig der aktuelle Monat (01-12) eingesetzt. Dieser Standardwert ist sinnvoll, wenn monacct am ersten Tage jedes Monats über cron ausgeführt werden soll. Monacct erstellt Summendateien in /usr/adm/acct/fiscal und startet Summendateien in /usr/adm/acct/sum neu.

Nulladm legt die angegebene Datei mit den Zugriffsrechten 664 an. Eigentümer und Gruppe dieser Datei ist adm. Es wird von verschiedenen Shell-Prozeduren für die Abrechnung aufgerufen.

Mit **prctmp** kann die Sessionsdatei (normalerweise /usr/adm/acct/nite/ctmp), die von acctcon1 erstellt wurde, ausgegeben werden.

Anhang 1: Administrator-Kommandos

Prdaily wird von runacct aufgerufen, um eine Auswertung der Abrechnungsdaten des vorangehenden Tages zu formatieren. Die Auswertung ist in `/usr/adm/acct/sum/rprtmmtt` abgelegt, wobei *MMTT* für den Monat und Tag der Auswertung steht. Sie können die aktuellen täglichen Abrechnungsauswertungen ausgeben, indem sie `prdaily` aufrufen. Unter der Option `-l` erhalten Sie Angaben über außergewöhnliche Nutzung einer Benutzer-Id des angegebenen Tages. Abrechnungsauswertungen vorangegangener Tage werden ausgegeben, indem durch *MMTT* das genaue Abrechnungsdatum spezifiziert wird. Frühere Tagesauswertungen werden bereinigt und sind daher nach einem Aufruf von `monacct` nicht mehr aufrufbar. Mit der Option `-c` erhalten Sie Informationen über außergewöhnliche Ressourcennutzung eines Kommandos. Dies kann jedoch nur für das aktuelle Datum abgefragt werden.

Mit **prtacct** können Sie jede gewünschte Summen-Abrechnungsdatei (`acct`) formatieren und ausgegeben.

Runacct bewirkt die Kumulierung der Abrechnungsdaten für Anschlußzeit, Prozesse, Gebühren und Plattenbelegung auf Tagesbasis. Ferner erstellt es Zusammenfassungen der Kommandobenutzung. Weitere Informationen siehe `runacct`.

Shutacct sollte während des ordnungsgemäßen Abschaltens des Systems (normalerweise in `/etc/shutdown`) aufgerufen werden, um die Prozeßabrechnung zu deaktivieren und einen Satz mit einem "*Grund*" in `/etc/wtmp` zu schreiben.

Startup sollte von `/etc/rc` aufgerufen werden, um die Abrechnung immer dann zu aktivieren, wenn das System hochgefahren wird.

Turnacct ist eine Schnittstelle zu `accton` (siehe `acct`), über die die Prozeßabrechnung aktiviert und deaktiviert werden kann. Das Argument `switch` deaktiviert die Abrechnung, benennt die aktuelle `/usr/adm/pacct` in den nächsten freien Namen in `/usr/adm/pacctincr` um (wobei *incr* eine Nummer ist, die mit 1 beginnt und bei jeder weiteren `pacct`-Datei um 1 erhöht wird) und aktiviert die Abrechnung sodann wieder. Diese Prozedur wird von `ckpacct` aufgerufen. Sie kann daher von `cron` verwaltet werden und dient dazu, daß `pacct` nicht zu groß wird.

Anhang 1: Administrator-Kommandos

DATEIEN:

/usr/adm/fee	Akkumulator für Gebühren.
/usr/adm/pacct	Aktuelle Datei für die Abrechnung pro Prozeß.
/usr/adm/pacct*	Wird verwendet, wenn pacct zu groß wird sowie während der Ausführung der täglichen Abrechnungsprozedur.
/etc/wtmp	Zusammenfassung der An- und Abmeldungen.
/usr/lib/acct/ptelus.awk	Außergewöhnliche Nutzung einer Login-ID.
/usr/lib/acct/ptecms.awk	Außergewöhnliche Nutzung eines Kommandos
/usr/adm/acct/nite	Arbeitsverzeichnis.
/usr/lib/acct	Enthält alle Administrator-Abrechnungskommandos.
/usr/adm/acct/sum	Verzeichnis, in dem die zusammengefaßten Abrechnungssummen enthalten sind. Dieses Verzeichnis sollte gesichert werden.

SIEHE AUCH:

Administrator-Kommandos: acct, acctcms, acctcon, acctmerge, acctprc, cron, diskusg, fwtmp, runacct
Dateiformate: acct, utmp
Benutzerkommando: acctcom
Systemschnittstelle: acct

Anhang 1: Administrator-Kommandos

brc, bcheckrc, rc, powerfail – Shell-Prozeduren für Systeminitialisierung

SYNTAX:

```
/etc/brc  
/etc/bcheckrc  
/etc/rc  
/etc/powerfail
```

BESCHREIBUNG:

Mit Ausnahme von **powerfail** werden diese Shell-Prozeduren über Einträge in `/etc/inittab` von `init` ausgeführt, wenn das System vom Single-User-Modus in einen anderen Run Level wechselt. **Powerfail** wird bei jedem erkannten Netzausfall ausgeführt.

Die Prozedur **brc** löscht die Datei `/etc/mnttab` (siehe Dateiformat `mnttab`), in der sich die Informationen über die eingehängten Dateisysteme befinden.

Bcheckrc führt alle notwendigen Konsistenzprüfungen durch, um das System für den Übergang in den Multi-User-Modus vorzubereiten. Die Prozedur verlangt das Setzen des Systemdatums und die Überprüfung der Dateisysteme mit `fsck`.

Rc startet alle System-Dämonen, bevor die Terminalleitungen für den Multi-User-Modus freigegeben werden. Darüber hinaus werden Dateisysteme eingehängt und die Abrechnung, die Fehlerprotokollierung, die Protokollierung der Systemaktivitäten und das Remote Job Entry (RJE) System in dieser Prozedur aktiviert.

Powerfail wird aufgerufen, wenn das System einen Netzausfall erkennt. Diese Prozedur hat hauptsächlich die Aufgabe, gegebenenfalls vorhandene programmierbare Mikroprozessoren wieder mit den entsprechenden Scripts zu laden. Ferner wird die Tatsache eines Netzausfalls protokolliert.

Anhang 1: Administrator-Kommandos

SIEHE AUCH:

Administrator-Kommandos: fsck, init
Dateiformate: inittab, mnttab

Anhang 1: Administrator-Kommandos

checkall – Schnelle Prüfprozedur für Dateisysteme

SYNTAX:

```
/etc/checkall
```

BESCHREIBUNG:

Bei der Prozedur checkall handelt es sich um einen Prototyp, der an die jeweiligen Verhältnisse angepaßt werden muß. Es folgt ein Beispiel:

```
# Root-Dateisystem selbst prüfen
fck /dev/nd00

# fsck für Laufwerke 0 und 1
dfck /dev/rdisk/0s[12345] - /dev/rdisk/1s1
```

In dem o. g. Beispiel (/dev/rdisk/1s1 320 K Blöcke und /dev/rdisk/0s[12345] jeweils maximal 65 K) würde ein sequentieller fsck 19 Minuten dauern. Die checkall-Prozedur dauert 11 Minuten.

Das Programm dfck gestattet Ihnen, mit zwei fsck-Programmen gleichzeitig einen Dialog zu führen. Um dies zu erleichtern, gibt dfck bei jeder Meldung den entsprechenden Dateisystemnamen mit an. Bei der Beantwortung der Fragen von fsck müssen Sie jeweils eine 1 oder 2 voranstellen (so daß die Antwort sich auf die erste bzw. zweite Dateisystemgruppe bezieht).

Wegen des bei doppelter Prüfung erforderlichen Lastausgleichs zwischen den Dateisystemen sollte das Kommando dfck immer über die Shell-Prozedur checkall ausgeführt werden.

Praktischerweise sind die Dateisysteme wie folgt eingeteilt:

```
dfck Dateisysteme-auf-Laufwerk-0 – Dateisysteme-auf-Laufwerk-1
dfck Dateisysteme-auf-Laufwerk-2 – Dateisysteme-auf-Laufwerk-3
```

...

Anhang 1: Administrator-Kommandos

Nachfolgend ein Beispiel für ein System mit drei Laufwerken (unter der Annahme von zwei großen Dateisystemen pro Laufwerk):

```
dfscck /dev/dsk/3s1 /dev/dsk/0s[14] - /dev/dsk/1s[14] /dev/dsk/3s4
```

Bitte beachten Sie, daß das erste Dateisystem auf Laufwerk 3 das erste in der Liste Dateisysteme-1 und das letzte in der Liste Dateisysteme-2 ist. Dadurch wird sichergestellt, daß Referenzen auf dieses Laufwerk sich zur Ausführungszeit nicht überlappen.

HINWEISE:

1. Dfscck darf **nicht** zum Prüfen des Root-Dateisystems verwendet werden.
2. Bei einer Prüfung, die eine Arbeitsdatei erfordert (siehe fsck Option -t) darf für die zwei Gruppen nicht dieselbe temporäre Datei verwendet werden (dadurch würden die Dateisysteme mit Sicherheit verstümmelt).
3. Die Prozedur dfscck ist nur dann sinnvoll, wenn im System mehrere physische E/A-Puffer eingerichtet sind.

SIEHE AUCH:

Administrator-Kommandos: dfscck, fsck

Anhang 1: Administrator-Kommandos

chroot – Wechseln des Wurzelverzeichnis für eine Kommandoausführung

SYNTAX:

/etc/chroot Wurzelverzeichnis Kommando

BESCHREIBUNG:

Das angegebene Kommando wird bezogen auf die neue Wurzel ausgeführt. Die Bedeutung des vorangestellten Schrägstriches (/) in Pfadnamen wird für ein Kommando und alle seine Abkömmlinge in das neue Wurzelverzeichnis geändert. Außerdem ist das anfängliche Arbeitsverzeichnis das neue Wurzelverzeichnis.

Beachten Sie aber bitte folgendes:

chroot Wurzelverzeichnis Kommando > Datei

erstellt die angegebene Datei relativ zur ursprünglichen und nicht zur neuen Wurzel.

Chroot kann nur vom Superuser ausgeführt werden.

Der Pfadname des neuen Wurzelverzeichnisses ist immer relativ zur aktuellen Wurzel: auch wenn ein chroot derzeit gilt, bezieht sich das Argument neue-Wurzel auf die aktuelle Wurzel des laufenden Prozesses.

HINWEIS:

Beim Ansprechen von Gerätedateien im Dateisystem der neuen Wurzel sollten Sie äußerst vorsichtig sein.

SIEHE AUCH:

Systemschnittstelle: chdir

Anhang 1: Administrator-Kommandos

clri – Löschen von I-Knoten

SYNTAX:

/etc/clri Dateisystem I-Nummer ...

BESCHREIBUNG:

Clri überschreibt die I-Knoten in dem angegebenen Dateisystem mit der angegebenen Nummer mit Nullen. Nach Ausführung eines clri werden alle Blöcke der betroffenen Dateien bei Ausführung des Kommandos fsck für das Dateisystem als „nicht vorhanden“ (missing) gemeldet.

Für die Gerätedatei, die mit dem angegebenen Dateisystem verbunden ist, wird Lese- und Schreiberlaubnis verlangt. Die I-Knoten können anschließend wieder zugeordnet werden.

Dieses Programm wird hauptsächlich verwendet, um eine Datei zu entfernen, die aus irgendeinem Grund in keinem Verzeichnis erscheint. Wird clri dazu benutzt, um einen I-Knoten zu entfernen, der noch in einem Verzeichnis vorhanden ist, muß der betreffende Eintrag herausgefunden und gelöscht werden. Wird nämlich der I-Knoten einer anderen Datei zugewiesen, verweist der alte Eintrag noch immer auf diese Datei. Das Löschen der alten Datei führt dann ebenfalls zur Zerstörung der neuen Datei. Der neue Eintrag verweist wiederum auf einen nicht zugeordneten I-Knoten, so daß die Gefahr besteht, daß dieser Kreislauf sich endlos wiederholt.

HINWEIS:

Für eröffnete Dateien kann clri nicht aufgerufen werden.

SIEHE AUCH:

Administrator-Kommandos: fsck, fsdb, ncheck
Dateiformat: fs

© Weitergabe sowie Vervielfältigung dieser Unterlage, Verwendung und Mitteilung ihres Inhalts nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zu Schadenersatz. Alle Rechte für den Fall der Patenterteilung oder Gebrauchsmustererteilung vorbehalten.

A1

Anhang 1: Administrator-Kommandos

cpset – Objektdateien in binären Verzeichnissen installieren

SYNTAX:

```
cpset [-o] Objektverzeichnis [Zugriffsmodus Eigentümer  
Gruppe]
```

BESCHREIBUNG:

Mit cpset wird die angegebene Objektdatei in dem angegebenen Verzeichnis installiert. Der Zugriffsmodus, der Eigentümer und die Gruppe der Zieldatei können in der Kommandozeile angegeben werden. Werden diese Angaben weggelassen, so sind zwei Ergebnisse möglich:

1. Hat der Benutzer von cpset administrative Rechte (d. h. wenn die numerische Benutzer-ID kleiner als 100 ist), werden die folgenden Standardwerte angeboten:

```
Zugriffsmodus – 0755  
Eigentümer – bin  
Gruppe – bin
```

2. Ist der Benutzer kein Administrator, entsprechen die Standardwerte für den Zugriffsmodus, den Eigentümer und die Gruppe der Zieldatei denjenigen des aufrufenden Benutzers.

Geben Sie die Option -o an, so hängt cpset das angegebene Objekt erst nach *OLDobjekt* im Zielverzeichnis um, bevor das neue Objekt installiert wird.

BEISPIELE:

```
cpset echo /bin 0755 bin bin  
cpset echo /bin  
cpset echo /bin/echo
```

Alle drei Beispiele haben dieselbe Wirkung, vorausgesetzt Sie haben die Rechte eines Administrators. Die Datei echo wird in /bin kopiert und erhält den Zugriffsmodus 0755. Eigentümer und Gruppe sind bin.

Anhang 1: Administrator-Kommandos

Aufgrund der Datei `/usr/src/destinations` stellt `cpset` das endgültige Ziel der Datei fest. Die Speicherortdatei enthält Paare von Pfadnamen, die durch Leerzeichen oder Tabs voneinander getrennt sind. Der erste Name ist das „offizielle“ Ziel (z. B.: `/bin/echo`). Der zweite Name ist das neue Ziel. Wird `echo` beispielsweise von `/bin` nach `/usr/bin` umgehängt, ist folgender Eintrag in `/usr/src/destinations` enthalten:

```
/bin/echo /usr/bin/echo
```

Wenn die eigentliche Installation erfolgt, prüft `cpset`, ob der „alte“ Pfadname nicht existiert. Existiert an dieser Stelle eine Datei, gibt `cpset` eine Warnmeldung aus und arbeitet weiter. Diese Datei existiert nicht auf einem Installationsband; es wird von `sites` verwendet, um lokal umgehängte Kommandos zu verfolgen. Die Prozeduren, mit denen die Quelle aufgebaut wird, sind zuständig für die Definition der „offiziellen“ Speicherstellen der Quelle.

Cross-Generierung

Mit der Umgebungsvariable `ROOT` wird die Zieldatei lokalisiert (in der Form `$ROOT/usr/src/destinations`). Dies ist notwendig, wenn auf einem Produktionssystem eine Cross-Generierung durchgeführt wird.

SIEHE AUCH:

Administrator-Kommando: `install`

Benutzerkommando: `make`

Anhang 1: Administrator-Kommandos

crash – Core-Images des Betriebssystems untersuchen

SYNTAX:

/etc/crash [System] [Namensliste]

BESCHREIBUNG:

Crash ist ein interaktives Dienstprogramm zum Untersuchen des Speicherabbildes (Core Image) des Betriebssystems. Es besitzt Einrichtungen zum Interpretieren und Formatieren der verschiedenen Kontrollstrukturen im System sowie weitere diverse Funktionen, die beim Untersuchen eines Dumps nützlich sind.

Die Argumente von crash sind der Name der Datei, in der das System-Image steht, und eine Datei mit einer Namensliste, die für Symbolwerte verwendet wird.

Die Standardwerte sind /dev/mem und /unix; daher kann crash ohne Argumente zum Untersuchen eines aktiven Systems verwendet werden. Wenn eine System-Image-Datei angegeben ist, wird davon ausgegangen, daß es sich um einen Core-Dump des Systems handelt, und der Prozeß, der zum Zeitpunkt des Systemabsturzes lief, wird als Standardprozeß angenommen. Dieser wird aufgrund eines Wertes, der vom Dump-Mechanismus auf einer bestimmten Adresse abgelegt wird, festgestellt.

KOMMANDOS:

Die Eingabe für crash hat im typischen Fall folgende Form:

Kommando [Optionen] [auszugebende Strukturen]

Wenn zulässig, beeinflussen die Optionen das Format der Ausgabe. Wenn keine spezifischen Strukturelemente angegeben sind, werden alle gültigen Einträge verwendet.

Anhang 1: Administrator-Kommandos

BEISPIEL:

```
proc - 12 15 3
```

gibt die Prozeßtabellenplätze 12, 15 und 3 in einem ausführlichen Format aus, während proc die gesamte Prozeßtabelle im Standardformat ausgeben würde.

Die Kommandos, die eine E/A mit Adressen ausführen, nehmen für 32-Bit-Maschinen Hexadezimalformat und für 16-Bit-Maschinen Oktalformat.

Derzeit gibt es folgende Kommandos:

user [*Liste mit Prozeßtabelleneinträgen*]

Aliasnamen: uarea, u_area, u.

Die Benutzerstruktur des angegebenen Prozesses wird entsprechend der im Prozeßtabelleneintrag enthaltenen Informationen ausgegeben. Wenn keine Eintragsnummer angegeben ist, werden die Informationen des letzten laufenden Prozesses ausgegeben. Geswapte Prozesse führen zu einer Fehlermeldung.

trace [-r] [*Liste mit Prozeßtabelleneinträgen*]

Aliasnamen: t.

Einen Kernel-Stack-Trace des aktuellen Prozesses erzeugen. Wenn die Option -r angegeben ist, beginnt der Trace bei dem in kfp gesicherten Stack Frame Pointer. Andernfalls beginnt der Trace beim untersten Element im Stack, und es wird versucht, gültige Stack Frames tiefer im Stack zu finden. Wird keine Eintragsnummer angegeben, so werden die Informationen vom letzten laufenden Prozeß ausgegeben.

kfp [*Stack Frame Pointer*]

Aliasnamen: r5, fp.

Ist kein Argument angegeben, wird der Wert ausgegeben, den das Programm für den Anfang des aktuellen Stack Frame hält (anfangs aus einer festen Adresse im Dump gesetzt) bzw. wird der Frame Pointer auf den angegebenen Wert gesetzt.

Anhang 1: Administrator-Kommandos

stack [*Liste mit Prozeßtabelleneinträgen*]

Aliasnamen: stk, s, kernel, k.

Einen Dump des Kernel-Stack eines Prozesses formatieren. Bei den ausgegebenen Adressen handelt es sich um virtuelle Systemdatenadressen und nicht um echte physikalische Adressen. Ist keine Eintragsnummer angegeben, so werden die Informationen vom letzten laufenden Prozeß ausgegeben.

proc [-[r]] [*Liste mit Prozeßtabelleneinträgen*]

Aliasnamen: ps, p.

Die Prozeßtabelle formatieren. Geben Sie die Option -r an, so werden nur die ablauffähigen Prozesse ausgegeben. Geben Sie nur - an, so wird eine längere Liste ausgegeben.

pcb [*Liste mit Prozeßtabelleneinträgen*]

Den Prozeßkontrollblock des aktuellen Prozesses ausgeben. Ist keine Eintragsnummer angegeben, so werden die Informationen vom letzten laufenden Prozeß ausgegeben.

i-node [-] [*Liste mit i-node-Tabelleneinträgen*]

Aliasnamen: ino, i.

Formatieren der I-Knoten-Tabelle. Geben Sie die Option - an, so werden auch die Datenblockadressen der I-Knoten ausgegeben.

file [*Liste mit Dateitabelleneinträgen*]

Aliasnamen: files, f.

Die Dateitabelle formatieren.

mount [*Liste mit mount-Tabelleneinträgen*]

Aliasnamen: mnt, m.

Die mount-Tabelle formatieren.

text [*Liste mit Texttabelleneinträgen*]

Aliasnamen: txt, x.

Die Texttabelle formatieren.

tty [*Typ*] [-] [*Liste mit Terminaleinträgen*]

Aliasnamen: term

Die tty-Strukturen ausgeben. Das Argument Typ bestimmt, welche Struktur verwendet wird (z. B. kl, dh, dz oder dzb auf DEC-Systemen; tn83, tn74 oder tn4 auf den Computern 3B 20S). Ein Standardwert für den Typ gibt es nicht. Ist jedoch einmal ein Typ angegeben worden, merkt sich das System diesen. Die Option - bewirkt, daß die Parameter des stty-Kommandos für die angegebene Zeile ausgegeben werden.

Anhang 1: Administrator-Kommandos

stat Bestimmte statistische Werte, die aus dem Dump hervorgehen, ausgeben. Hierzu gehört die sogenannte Panikmeldung (wenn eine schwerwiegende Störung aufgetreten ist), der Zeitpunkt des Absturzes, der Systemname und die vom Dump-Mechanismus im untersten Hauptspeicherbereich gesicherten Register.

var Aliasnamen: tunables, tunable, tune, v.
Die einstellbaren Systemparameter ausgeben.

buf [*Liste mit Puffer-Headern*]
Aliasnamen: hdr, bufhdr.
Die Systempuffer-Header formatieren.

buffer [*Format*] [*Liste mit Puffern*]
Aliasname: b.
Die Daten in einem Systempuffer entsprechend dem angegebenen Format ausgeben. Ist kein Format angegeben, so wird das vorangegangene Format verwendet. Gültige Formate sind decimal, octal, hex, character, byte, directory, i-node und write. Bei dem letztgenannten Format wird eine Datei im aktuellen Verzeichnis (siehe DATEIEN) angelegt, die die Pufferdaten enthält.

callout
Aliasnamen: calls, call, c, timeout, time, tout.
Alle Einträge in der callout-Tabelle ausgeben.

map [*Liste mit Map-Namen*]
Die genannten System-Map-Strukturen formatieren.

nm [*Liste mit Symbolen*]
Den in der Namenslistendatei gefundenen Symbolwert und Typ ausgeben.

ts [*Liste mit Textadressen*]
Die nächstliegenden Textsymbole für die angegebenen Adressen suchen.

ds [*Liste mit Datenadressen*]
Die nächstliegenden Datensymbole für die angegebenen Adressen suchen.

Anhang 1: Administrator-Kommandos

od [*Symbolname* oder *Adresse*] [*Zähler*] [*Format*]

Aliasnamen: dump, rd.

Zählerdaten entsprechend dem angegebenen Format ab dem angegebenen Symbolnamen bzw. der Adresse dumpen. Mögliche Formate sind octal, longoct, decimal, longdec, character, hex und byte.

! Aufrufen der Shell.

q crash verlassen.

? Überblick der Kommandos ausgeben.

ALIASNAMEN:

Neben den Aliasnamen für die Kommandos gibt es für zahlreiche Formate Aliasnamen. Es folgen einige davon:

byte	b.
character	char, c.
decimal	dec, e.
directory	direct, dir, d.
hexadecimal	hexadec, hex, h, x.
i-node	ino, i.
longoct	lo, O.
octal	oct, o.
write	w.

DATEIEN:

/usr/include/sys/*.h	Header-Dateien für Tabellen- und Strukturinformationen
/dev/mem	Core-Image-Datei des Systems (Standard)
/unix	Standardmäßige Namenslistendatei
buf.#	Angelegte Dateien mit Pufferdaten

Anhang 1: Administrator-Kommandos

HINWEIS:

Ein Stack-Trace für den aktuellen Prozeß auf einem laufenden System funktioniert nicht.

SIEHE AUCH:

Administrator-Kommando: mount

Benutzerkommandos: nm, ps, sh, stty

Anhang 1: Administrator-Kommandos

cron – Ständige Überprüfung von /usr/lib/crontab und Ausführung der darin enthaltenen Instruktionen

SYNTAX:

`/etc/cron`

BESCHREIBUNG:

Cron führt Kommandos an vorgegebenen Tagen und Uhrzeiten entsprechend den in den crontab-Dateien enthaltenen Anweisungen aus. Systembenutzer können mit Hilfe des crontab-Kommandos ihre eigenen crontab-Dateien anlegen. Kommandos, die nur einmal ausgeführt werden sollen, sollten durch `at` aufgerufen werden.

Da cron während der Laufzeit des Systems nicht terminiert, sollte es nur einmal aufgerufen werden. Sinnvollerweise wird cron durch den Initialisierungsprozeß über die Datei `/etc/rc` (siehe `init`) aufgerufen.

Cron überprüft crontab-Dateien und `at`-Kommandos nur während der Prozeß-Initialisierung oder wenn eine Datei geändert wird. Dadurch wird der CPU-Overhead – gegenüber einer regelmäßigen Prüfung – deutlich reduziert.

Jede Zeile der Datei crontab besteht aus sechs Feldern, die durch Blanks oder Tabs getrennt sind. Die ersten fünf Felder enthalten Ziffern in folgender Reihenfolge:

1. Feld: Minute (0–59)
2. Feld: Stunde (0–23)
3. Feld: Tag des Monats (1–31)
4. Feld: Monat des Jahres (1–12)
5. Feld: Wochentag (0–6, 0=Sonntag)

Jedes dieser Felder kann folgendermaßen aufgebaut sein:

- Eine Zahl in der o. g. Form.
- Zwei Zahlen, durch Bindestrich getrennt (inklusive Zeitraumangabe).

Anhang 1: Administrator-Kommandos

- Liste von Zahlen, getrennt durch Kommata (jede der angegebenen Zahlen wird interpretiert).
- Ein Stern (alle möglichen legalen Werte werden interpretiert).

Das sechste Feld enthält die Angaben für die Shell, was zu der spezifizierten Zeit auszuführen ist. Ein %-Zeichen in diesem Feld wird als Zeilenende-Zeichen genommen. Nur die erste Zeile (bis zum %-Zeichen oder Zeilenende-Zeichen) wird als Kommandozeile interpretiert. Die folgenden Zeilen sind für die Shell Standardeingabe.

DATEIEN:

/usr/lib/cron	Verzeichnis für cron-Dateien
/usr/adm/cron/log	Accounting-Informationen
/usr/spool/cron	Spooling

MELDUNGEN:

Ein Protokoll aller von cron durchgeführten Aktionen wird in /usr/lib/cron/log geführt.

SIEHE AUCH:

Benutzerkommandos: at, crontab, sh

Anhang 1: Administrator-Kommandos

dcopy – Dateisysteme reorganisieren

SYNTAX:

```
/etc/dcopy [-sX][-an][-d][-v][-fGröße[:l-Größe]] Dateisystem1
Dateisystem2
```

BESCHREIBUNG:

Das Kommando dcopy dient dazu, die Zugriffszeit auf Dateisysteme zu optimieren. Dcopy kopiert das Dateisystem1 nach Dateisystem2. *Dateisystem1* ist der Name des vorhandenen Dateisystems; *Dateisystem2* ist ein entsprechend dimensioniertes Dateisystem, das das reorganisierte Dateisystem aufnehmen soll. Um optimale Ergebnisse zu erzielen, sollte *Dateisystem1* das Raw Device und *Dateisystem2* das blockorientierte Gerät sein. Dcopy sollte nur für ausgehängte Dateisysteme ausgeführt werden (im Falle des Rootdateisystems ist dies auf eine neue Platte zu kopieren). Ohne Argumente kopiert dcopy Dateien aus *Dateisystem1* und komprimiert dabei die Verzeichnisse, in denen leere Einträge entfernt werden; gleichzeitig ordnet es aufeinanderfolgende Blöcke in einer Datei so an, daß sich ein optimaler Plattenzugriff ergibt. Folgende Optionen sind möglich:

- sX Geräteinformationen zur Optimierung der Blockorganisation in einer Datei bereitstellen. X kann dieselben Werte annehmen wie die Option -s des Administrator-Kommandos fsck.
- an Die Dateien, die seit *n* Tagen nicht mehr angesprochen wurden, hinter den freien Blöcken des Zieldateisystems plazieren (Standardwert für *n* ist 7). Ist *n* nicht angegeben, werden die Dateien nicht versetzt.
- d Die Reihenfolge der Verzeichniseinträge unverändert lassen (standardmäßig werden Unterverzeichnisse an den Anfang der Verzeichnisse versetzt).
- v Aktuelle Meldungen, wie viele Dateien verarbeitet wurden und wie groß die Listen der freien Blöcke der Quelle und des Ziels sind.
- fGröße[:l-Größe]
Die Größe von *Dateisystem2* und der l-Knoten-Liste. Geben Sie diese Option (oder :l-Größe) nicht an, werden die Werte von *Dateisystem1* verwendet.

Anhang 1: Administrator-Kommandos

Dcopy empfängt Interrupts und Quit-Signale und gibt über seinen Verlauf Auskunft. Um dcopy zu beenden, geben Sie ein Quit-Signal ein.

SIEHE AUCH:

Administrator-Kommandos: fsck, mkfs

Benutzerkommando: ps

Anhang 1: Administrator-Kommandos

devnm – Gerätenamen ermitteln

SYNTAX:

```
/etc/devnm [Namen]
```

BESCHREIBUNG:

Devnm dient zur Anzeige der logischen Platte, welche das Dateisystem mit dem angegebenen Namen enthält.

Namen müssen Sie immer als vollen Pfadnamen angeben.

BEISPIEL:

```
/etc/devnm /u2/aria
```

zeigt an:

```
nd04 /u2/aria
```

wenn /u2/aria der logischen Platte /dev/nd04 zugeordnet ist.

DATEIEN:

```
/dev/nd0*
```

```
/etc/mnttab
```

SIEHE AUCH:

Administrator-Kommandos: brc, setmnt

Anhang 1: Administrator-Kommandos

df – Anzahl der freien Plattenblöcke melden

SYNTAX:

```
df [-t] [-f] [Dateisysteme]
```

BESCHREIBUNG:

Df gibt die Anzahl der freien Blöcke und der freien I-Knoten aus, die in den Dateisystemen noch verfügbar sind. Zum Zeitpunkt der Auswertung müssen die Dateisysteme gemounted sein. DF wertet die in den Superblöcken geführten Zähler aus. Die Dateisysteme können entweder mit ihrem Gerätenamen (z. B. /dev/nd04) oder mit ihrem Wurzelverzeichnisnamen (z. B. /u2) angegeben wrden. Geben Sie kein Argument an, wird der freie Platz in allen eingehängten Dateisystemen ausgegeben.

Optionen:

- t Die Anzahl der allokierten Blöcke wird ebenfalls ausgegeben.
- f Es wird nur die aktuelle Anzahl der Blöcke ausgegeben, die in der „Free List“ enthalten sind (freie I-Knoten werden nicht angegeben). Mit dieser Option werden „Raw-Devices“ ausgewertet.

DATEIEN:

/dev/nd0*

/etc/mnttab

SIEHE AUCH:

Dateiformate: fs, mnttab

Anhang 1: Administrator-Kommandos

diskusg – Plattenabrechnungsdaten nach Benutzer-ID erzeugen

SYNTAX:

```
diskusg [Optionen] [Dateien]
```

BESCHREIBUNG:

Diskusg erzeugt einen Zwischenstand der Plattenabrechnungsinformationen aus den angegebenen Dateien bzw. aus der Standardeingabe. Diskusg gibt auf der Standardausgabe eine Zeile pro Benutzer im folgenden Format aus:

```
uid login #Blöcke
```

Bedeutung:

uid Die numerische Benutzer-ID des Benutzers.

login Login-Name des Benutzers.

#Blöcke Die Gesamtzahl der diesem Benutzer zugeordneten Plattenblöcke.

Diskusg liest für die Plattenabrechnung in der Regel nur die I-Knoten der Dateisysteme. In diesem Fall steht *Dateien* für die Gerätedateinamen dieser Geräte.

Diskusg erkennt die folgenden Optionen:

- s Die Eingabedaten befinden sich bereits im Ausgabeformat für diskusg. Diskusg faßt alle Zeilen für einen Benutzer zu einer einzelnen Zeile zusammen.
- v Ausführliche (verbose) Ausgabe. Auf der Standardfehlerausgabe werden alle Dateien ausgegeben, die niemandem zugerechnet werden.
- i *Liste* Die Daten in den Dateisystemen ignorieren, deren Dateisystemname in *Liste* erscheint. Dies ist eine Liste von Dateisystemnamen, die durch Kommata voneinander getrennt oder in Anführungszeichen eingeschlossen sind. Diskusg vergleicht alle Namen in dieser Liste mit dem im Volume-ID eingetragenen Dateisystemnamen (siehe Administrator-Kommando labelit).

Anhang 1: Administrator-Kommandos

- p *Datei* *Datei* als Name für die Paßwortdatei zur Erzeugung von Login-Namen verwenden. Standardmäßig wird /etc/passwd verwendet.
- u *Datei* Sätze für Dateien, die niemandem zugerechnet werden, in die angegebene Datei schreiben. Die Sätze enthalten den Gerätedateinamen, die I-Knoten-Nummer und die Benutzer-ID.

Die Ausgabe von diskusg ist normalerweise die Eingabe von acctdisk (acct). Dieses Kommando erzeugt Summenabrechnungssätze, die mit anderen Abrechnungssätzen kombiniert und gemischt werden können. Diskusg wird in der Regel in dodisk ausgeführt (siehe acctsh).

BEISPIELE:

Durch folgendes Script werden tägliche Plattenabrechnungsinformationen erzeugt:

```
for i in /dev/rp00 /dev/rp01 /dev/rp10 /dev/rp11; do
    diskusg $i > dtmp.'basename $i' &
done
wait
diskusg -s dtmp.* | sort +0n +1 | acctdisk > diskacct
```

DATEIEN:

/etc/passwd dient zur Konvertierung der Benutzer-ID in Login-Namen

SIEHE AUCH:

Administrator-Kommandos: acct, acctsh
Dateiformat: acct

Anhang 1: Administrator-Kommandos

errdead – Fehlersätze aus einem Speicher-Dump extrahieren

SYNTAX:

```
/etc/errdead [Dumpdatei] [Namensliste]
```

BESCHREIBUNG:

Erkennt das System Hardware-Fehler, wird ein Fehlersatz erzeugt, der Angaben über den Fehler enthält. Ist *errdemon* nicht aktiv oder stürzt das System ab, bevor der Satz in die Fehlerdatei geschrieben werden kann, hält das System die Fehlerinformation in einem lokalen Puffer. *Errdead* untersucht einen System-Dump (oder den Hauptspeicher), holt sich die entsprechenden Fehlersätze heraus und übergibt sie zur Auswertung an *errpt*.

Dumpdatei gibt die Datei (oder den Speicher) an, die/der untersucht werden soll. *Namensliste* ist die System-Namensliste; ist dieser Parameter nicht angegeben, wird standardmäßig */unix* angenommen.

DATEIEN:

<i>/unix</i>	System-Namensliste
<i>/usr/bin/errpt</i>	Auswertungsprogramm
<i>/usr/tmp/errXXX</i>	Temporäre Datei

DIAGNOSE:

Diagnosemeldungen können von *errdead* oder *errpt* kommen. In beiden Fällen sind diese Meldungen aber selbsterklärend.

SIEHE AUCH:

Administrator-Kommandos: *errdemon*, *errpt*

Anhang 1: Administrator-Kommandos

errdemon – Fehlerprotokollierungs-Dämon

SYNTAX:

`/usr/lib/errdemon [Datei]`

BESCHREIBUNG:

Der Fehlerprotokollierungs-Dämon `errdemon` übernimmt Fehlersätze vom Betriebssystem, indem er die Gerätedatei `/dev/error` liest. Diese Fehlersätze werden in die angegebene Datei geschrieben. Wird bei der Aktivierung des Dämons keine Datei angegeben, so wird `/usr/adm/errfile` beschrieben. *Datei* wird angelegt, wenn sie noch nicht vorhanden ist; ist sie bereits vorhanden, werden die Fehlersätze an das Dateiende angefügt, so daß keine vorhandenen Fehlerdaten verlorengehen. `Errdemon` wertet die Fehlersätze nicht aus. Für diese Auswertung ist `errpt` zuständig. Der Fehlerprotokollierungs-Dämon terminiert aufgrund von `kill`-Signalen (siehe Benutzerkommando `kill`). Nur der Superuser kann den Dämon starten. Zu einem Zeitpunkt kann immer nur ein Dämon aktiv sein.

DATEIEN:

<code>/dev/error</code>	Quelle der Fehlersätze
<code>/usr/adm/errfile</code>	Standard-Fehlerdatei

SIEHE AUCH:

Administrator-Kommandos: `errpt`, `errstop`
 Spezielle Datei: `err`
 Benutzerkommando: `kill`

Anhang 1: Administrator-Kommandos

errpt – Auswertung protokollierter Fehler

SYNTAX:

`errpt [Optionen] [Dateien]`

BESCHREIBUNG:

Errpt verarbeitet die von `errdemon` erfaßten Daten und erzeugt eine Auswertung dieser Daten. Die standardmäßige Auswertung besteht aus einer Zusammenfassung aller Fehler, die in den angegebenen Dateien verzeichnet sind. Die Optionen gelten für alle Dateien und sind weiter unten beschrieben. Ist keine Datei angegeben, versucht `errpt` `/usr/adm/errfile` zu lesen.

In der zusammenfassenden Auswertung werden die Optionen, die die Vollständigkeit einschränken, die Zeitangabe des frühesten und spätesten aufgetretenen Fehlers und die Gesamtzahl der Fehler einer oder mehrerer Arten ausgewiesen. In jeder Zusammenfassung für ein Gerät werden die Gesamtzahl der nicht behobenen Fehler, der behobenen Fehler, der nicht protokollierbaren Fehler, der E/A-Operationen an dem Gerät und die sonstigen Aktivitäten, die am Gerät stattgefunden haben, verzeichnet. Die Fälle, in denen `errpt` Eingabedaten nicht ohne weiteres lesen konnte, werden als Lesefehler gezählt.

Jede detaillierte Auswertung enthält neben den spezifischen Fehlerinformationen alle Fälle, in denen der Fehlerprotokollierungs-Prozeß gestartet und gestoppt wurde sowie alle Zeitwechsel (über das Benutzerkommando `date`), die im Auswertungszeitraum stattgefunden haben. An eine detaillierte Auswertung schließen sich immer die Summen für jeden in der Auswertung enthaltenen Fehlertyp an.

Eine Auswertung kann mit Hilfe folgender Optionen auf bestimmte Sätze beschränkt werden:

- s *Datum* Alle Sätze ignorieren, die vor dem angegebenen Datum eingetragen wurden, wobei Datumsangabe die Form `MMTTHHMMJJ` hat, entsprechend dem Kommando `date`.
- e *Datum* Alle Sätze ignorieren, die nach *Datum* eingetragen wurden. Datum muß in der o. g. Form angegeben werden.

Anhang 1: Administrator-Kommandos

- a Eine detaillierte Auswertung mit sämtlichen Fehlertypen wird erstellt.
- d *Geräteliste*
Die detaillierte Auswertung wird auf Daten über die in *Geräteliste* angegebenen Geräte beschränkt, wobei die Geräteliste in zweierlei Form angegeben werden kann: eine Liste von Geräte-IDs, die durch Kommata voneinander getrennt sind, oder eine Liste von Geräte-IDs, die in Anführungszeichen eingeschlossen und durch ein Komma und/oder mehrere Leerzeichen voneinander getrennt sind. Bei den Geräten, für die Fehler protokolliert werden, handelt es sich um iocdisk und ioctape.
- p *n* Der Umfang einer detaillierten Auswertung wird auf *n* Seiten beschränkt.
- f In einer detaillierten Auswertung sollen für blockorientierte Geräte nur die nicht behobenen Fehler gemeldet werden.

DATEIEN:

/usr/adm/errfile Standard-Fehlerdatei

SIEHE AUCH:

Administrator-Kommandos: errdead, errdemon

Dateiformat: errfile

Benutzerkommando: date

Anhang 1: Administrator-Kommandos

errstop – Beenden des Fehlerprotokollierungs-Dämons

SYNTAX:

`/etc/errstop [Namensliste]`

BESCHREIBUNG:

Der Fehlerprotokollierungs-Dämon `errdemon` wird durch `errstop` beendet. Dazu wird zunächst das Benutzerkommando `ps` aufgerufen, um die Identität des Dämons festzustellen, und dann wird ein softwaremäßiges Kill-Signal (siehe Systemaufruf `signal`) an ihn abgesetzt. Ist keine Namensliste angegeben, so wird `/unix` als System-Namensliste verwendet. Nur der Superuser kann `errstop` anwenden.

DATEIEN:

`/unix` System-Namensliste

SIEHE AUCH:

Administrator-Kommando: `errdemon`

Benutzerkommando: `ps`

Systemschnittstellen: `kill`, `signal`

Anhang 1: Administrator-Kommandos

ff – Dateinamen und Statistiken für ein Dateisystem auflisten

SYNTAX:

/etc/ff [Optionen] Gerätedatei

BESCHREIBUNG:

ff liest die I-Liste und die Verzeichnisse von *Gerätedatei*, wobei es davon ausgeht, daß es sich um ein Dateisystem handelt. Es sichert die I-Knoten-Daten für Dateien, die die Auswahlkriterien erfüllen. Die Ausgabe besteht aus dem Pfadnamen für jeden gesicherten I-Knoten und allen weiteren Dateiinformationen, die mit Hilfe der unten aufgeführten Ausgabeoptionen angefordert wurden. Die Ausgabefelder sind positionsgebunden. Die Ausgabe erfolgt in der Reihenfolge der I-Knoten. Die Felder werden durch Tabs getrennt. Standardmäßig gibt ff Zeilen in der folgenden Form aus:

Pfadname I-Nummer

Sind alle Optionen aktiv, werden folgende Felder ausgegeben:

Pfadname I-Nummer Größe Benutzer-ID

Das Argument *n* in den folgenden Optionsbeschreibungen steht für eine Dezimalzahl (wahlweise mit Vorzeichen), wobei *+n* für mehr als *n*, *-n* für weniger als *n* und *n* für genau *n* steht. Ein Tag ist als Zeitraum von 24 Stunden definiert.

Optionen:

- i Die I-Knoten-Nummer wird nicht hinter jedem Pfadnamen ausgegeben.
- l Es wird eine ergänzende Liste aller Pfadnamen für Dateien mit mehreren Verknüpfungen (Links) erzeugt.
- p *Präfix* Das angegebene Präfix wird jedem erzeugten Pfadnamen vorangestellt. Standardwert ist der Punkt (.).
- s Hinter jedem Pfadnamen wird die Dateigröße in Bytes ausgegeben.

Anhang 1: Administrator-Kommandos

- u Hinter jedem Pfadnamen wird der Login-Name des Eigentümers ausgegeben.
- a *n* Auswählen, wenn der I-Knoten in den letzten *n* Tagen angesprochen wurde.
- m *n* Auswählen, wenn der I-Knoten in den letzten *n* Tagen modifiziert wurde.
- c *n* Auswählen, wenn der I-Knoten in den letzten *n* Tagen geändert wurde.
- n *Datei* Auswählen, wenn der I-Knoten später als die im Argument angegebene Datei modifiziert wurde.
- i *I-Knoten-Liste*
 Namen nur für die in der Liste angegebenen I-Knoten generieren.

BEISPIELE:

Eine Liste der Namen aller Dateien in einem angegebenen Dateisystem kann wie folgt erzeugt werden:

```
ff -l /dev/diskroot
```

Ein Verzeichnis der Dateien und I-Nummern, die in einem Dateisystem enthalten sind und in den letzten 24 Stunden modifiziert wurden, wird wie folgt erstellt:

```
ff -m -1 /dev/diskusr > /log/incbackup/usr/tuesday
```

Die Pfadnamen für die I-Knoten 451 und 76 in einem angegebenen Dateisystem können wie folgt festgestellt werden:

```
ff -i 451,76 /dev/rtp7
```

Anhang 1: Administrator-Kommandos

HINWEISE:

Für einen mehrfach verknüpften I-Knoten wird nur ein einziger der möglichen Pfadnamen generiert, außer wenn die Option -I angegeben ist. Ist -I spezifiziert, so gelten für die generierten Namen keine Auswahlkriterien. Alle möglichen Namen für alle verknüpften Dateien in dem Dateisystem werden ausgegeben.

Bei sehr großen Dateisystemen reicht der Hauptspeicher für die Ausführung von ff möglicherweise nicht aus.

SIEHE AUCH:

Administrator-Kommandos: findc, frec, ncheck
Benutzerkommando: find

Anhang 1: Administrator-Kommandos

filesave, tapesave – Tägliche/wöchentliche Datensicherung der Dateisysteme eines UNIX-Systems

SYNTAX:

/etc/filesave.?

/etc/tapesave

BESCHREIBUNG:

Diese Shell-Prozeduren dienen als Modelle. Sie stellen eine einfache interaktive Bedienerumgebung für die Sicherung von Dateisystemen zur Verfügung. Filesave.? ist für die tägliche Sicherung von Platte zu Platte und tapesave für die wöchentliche Sicherung von Platte auf Band vorgesehen.

Mit Hilfe des Suffixes .? kann ein anderes System angegeben werden, wenn zwei (oder mehr) Maschinen Plattenlaufwerke (oder Bandlaufwerke) gemeinsam benutzen oder eins der Systeme für die Datensicherung beider Systeme benutzt wird.

SIEHE AUCH:

Administrator-Kommandos: shutdown, volcopy

Anhang 1: Administrator-Kommandos

finc – Schnelle inkrementelle Datensicherung auf Band

SYNTAX:

finc [Auswahlkriterien] Dateisystem Raw-Band

BESCHREIBUNG:

Finc kopiert das *Dateisystem* (Eingabe) selektiv auf das angegebene Raw-Band (Ausgabe). Die Vorsicht gebietet, daß das Eingabe-Dateisystem schreibgeschützt montiert ist, um eine absolut korrekte Datensicherung zu gewährleisten. Aber auch im Schreib-Lese-Modus kann die Datensicherung akzeptabel sein. Das Band muß zuvor über labelit (siehe volcopy) mit einem Kennsatz versehen werden. Die Auswahl der zu kopierenden Teile wird durch die Auswahlkriterien gesteuert, wobei nur diejenigen I-Knoten/Dateien ausgewählt werden, die die Bedingungen erfüllen.

Es ist ratsam, vor der Datensicherung mit *finc* das Kommando *ff* auszuführen und die Ausgabe von *ff* als Index für den Bandinhalt aufzubewahren. Dateien auf einem mit *finc* erstellten Band können mit dem Kommando *frec* wieder eingespielt werden.

Das Argument *n* in den folgenden Auswahlkriterien steht für eine Dezimalzahl (wahlweise mit Vorzeichen), wobei *+n* für mehr als *n*, *-n* für weniger als *n* und *n* für genau *n* steht. Ein Tag ist als Zeitraum von 24 Stunden definiert.

Auswahlkriterien:

- a *n* Bedingung ist erfüllt, wenn die Datei in den letzten *n* Tagen angesprochen wurde.
- m *n* Bedingung ist erfüllt, wenn die Datei in den letzten *n* Tagen modifiziert wurde.
- c *n* Bedingung ist erfüllt, wenn der I-Knoten in den letzten *n* Tagen geändert wurde.
- r *Datei* Bedingung ist bei jeder Datei erfüllt, die später modifiziert wurde als die im Argument angegebene Datei.

Anhang 1: Administrator-Kommandos

BEISPIEL:

Um alle Dateien, die in den letzten 48 Stunden modifiziert wurden, auf ein Band zu sichern, geben Sie bitte folgenden Befehl an.

```
finc -m -2 /dev/rdiskusr /dev/rtp0
```

SIEHE AUCH:

Administrator-Kommandos: ff, frec, volcopy

Benutzerkommando: cpio

Anhang 1: Administrator-Kommandos

frec – Dateien von einem Sicherungsband rekonstruieren

SYNTAX:

/etc/frec [-p Pfad] [-f Auftragsdatei] Band I-Nummer:Name ...

BESCHREIBUNG:

Frec rekonstruiert Dateien von dem angegebenen Sicherungsband, die mit volcopy oder finc ausgelagert wurden. Dazu müssen die entsprechenden I-Nummern angegeben werden. Die rekonstruierten Daten werden in der Datei *Name* abgelegt.

Mit der Option *-p* können Sie ein Standard-Pfadpräfix angeben, das von dem aktuellen Arbeitsverzeichnis abweicht. Dieses Präfix wird allen Namen vorangestellt, die nicht vollständig explizit angegeben sind, d. h. die nicht mit */* oder *./* beginnen. Fehlen bei der Rekonstruktion von Dateien Verzeichnisse, werden sie angelegt.

Optionen:

- p Pfad* Gibt einen Präfix-Pfad an, mit dem alle Namen, die nicht mit */* oder *./* beginnen, ergänzt werden.
- f Datei* Frec entnimmt die Angaben über I-Nummern und Namen der angegebenen Datei. Das Format dieser Datei ist ebenfalls *I-Nummer:Name*. Diese Angaben stehen jeweils in einer Zeile.

BEISPIELE:

Um eine Datei zu rekonstruieren, die mit der I-Nummer 2547 gesichert wurde, geben Sie bitte folgendes Kommando an:

frec /dev/rmt0 2547:doris

Diese Datei wird dann unter dem Namen *doris* in Ihrem aktuellen Verzeichnis abgelegt.

Anhang 1: Administrator-Kommandos

Die gesicherten Dateien mit den I-Nummern 14156, 1232 und 3141 werden wie folgt in die Dateien /usr/src/cmd/a, /usr/src/cmd/b und /usr/doris/a.c eingespielt:

```
frec -p /usr/src/cmd /dev/rmt0 14156:a 1232:b 3141:/usr/doris/a.c
```

SIEHE AUCH:

Administrator-Kommandos: ff, find, volcopy

Benutzerkommando: cpio

Anhang 1: Administrator-Kommandos

fsck, dfscck – Konsistenzprüfungen eines Dateisystems

SYNTAX:

```

/etc/fsck  [-y] [-n] [-sX] [-SX] [-t Datei] [-q] [-f] [-D] [Gerä-
tedatei ...]

/etc/dfscck [Optionen1] Gerätedatei1 - [F2Optionen2] Gerä-
tedatei2

```

BESCHREIBUNG:

Fsck überprüft und repariert inkonsistente Dateisysteme interaktiv. Werden bei dem Prüflauf keine Inkonsistenzen gefunden, gibt fsck die Anzahl der Dateien, Anzahl der belegten Blöcke und Anzahl der freien Blöcke aus. Ist das Dateisystem inkonsistent, wird der Bediener jedesmal um seine Zustimmung gebeten, bevor eine Korrektur versucht wird. Hier sei ausdrücklich darauf hingewiesen, daß bei den meisten Korrekturmaßnahmen Daten verlorengehen. Den Umfang und die Bedeutung des Datenverlustes kann man anhand der Diagnoseausgabe feststellen. Vor jeder Korrektur wartet das System standardmäßig, bis der Bediener mit ja oder nein antwortet. Hat der Bediener keine Schreibberechtigung, führt fsck standardmäßig den Prüflauf entsprechend der Option -n aus.

Optionen:

- y Alle fsck-Fragen werden automatisch mit yes beantwortet.
- n Alle fsck-Fragen werden automatisch mit no beantwortet. Das Dateisystem ist nicht zum Schreiben eröffnet.
- sX Ignorieren der aktuellen freien Liste und unbedingte Konstruktion einer neuen freien Liste durch Beschreiben des Superblocks in dem Dateisystem. Das Dateisystems sollte während dieser Zeit ausgehängt sein. Ist dies nicht möglich, muß das System sofort nach Beendigung der Ausführung gebootet werden. Dies ist notwendig, damit die alte Kopie des Superblocks nicht mehr benutzt wird.

Anhang 1: Administrator-Kommandos

Diese Option unterstützt die optimale Organisation der freien Liste. Folgende Angaben von *X* werden auf den angegebenen Geräten unterstützt:

-s3 (RP03)

-s4 (RP04, RP05, RP06)

-s(Blocks-per-cylinder:Blocks-to-skip)
alles andere

Geben Sie keinen Wert für *X* an, werden die ursprünglichen Werte des Dateisystems benutzt. Sind diese Werte nicht spezifiziert, ist 400:7 maßgebend.

- SX Bedingtes Rekonstruieren der freien Liste. Diese Option verhält sich wie -sX, mit dem Unterschied, daß die freie Liste nur neu aufgebaut wird, wenn keine Unstimmigkeiten im Dateisystem entdeckt wurden.
- t Hat fsck nicht genügend Speicherplatz zur Verfügung, um seine Tabellen zu halten, benutzt es eine Scratch-Datei. Geben Sie die Option -t an, wird das nächste Argument als Name der Scratch-Datei genommen. Ohne Angabe dieser Option fragt fsck den Bediener nach einem Dateinamen, wenn eine Scratch-Datei benötigt wird. Die angegebene Datei darf nicht in dem zu prüfenden Dateisystem liegen. Ist sie keine Spezialdatei oder ist sie noch nicht angelegt, wird sie nach Beendigung von fsck gelöscht.
- q Keine Ausgabe von Nachrichten in Phase 1. Nicht-referenzierte Fifos werden gelöscht. Alle fsck-Reparaturvorschläge werden automatisch durchgeführt.
- D Die Verzeichnishierarchie wird auf gültige Verzeichniseinträge und auf korrekte Verkettung geprüft.
- f Nur die Überprüfungen in den Phasen 1 und 5 werden durchgeführt (fast check). Falls nötig, wird die freie Liste rekonstruiert (Phase 6).

Geben Sie keine Dateisysteme an, werden die Namen der zu prüfenden Dateisysteme der Datei /etc/checklist entnommen.

Anhang 1: Administrator-Kommandos

Inkonsistenzen werden wie folgt geprüft:

1. Blöcke, die von mehr als einem I-Knoten oder der Liste freier Blöcke beansprucht werden.
2. Blöcke, die von einem I-Knoten oder der Liste der freien Blöcke außerhalb des Bereichs des Dateisystems beansprucht werden.
3. Falscher Link-Zähler.
4. Größenprüfungen:
Unkorrekte Anzahl von Blöcken.
Verzeichnisgröße hat nicht das richtige Format.
5. Falsches I-Knoten-Format.
6. Blöcke, die nirgends verzeichnet sind.
7. Verzeichnisprüfungen:
Datei verweist auf nicht zugeordneten I-Knoten.
I-Knoten-Nummer außerhalb des zulässigen Bereichs.
8. Superblock-Prüfungen:
Mehr als 65536 I-Knoten.
Mehr Blöcke für I-Knoten, als im Dateisystem vorhanden sind.
9. Fehlerhaftes Format der Liste freier Blöcke.
10. Zähler der Summe freier Blöcke und/oder der freien I-Knoten falsch.

Verwaiste Dateien und Verzeichnisse (die zugeordnet sind aber nicht referenziert werden) werden mit Zustimmung des Bedieners neu verknüpft und im Verzeichnis lost+found untergebracht. Als Name wird die I-Knoten-Nummer zugeordnet. Einzige Voraussetzung ist, daß das Verzeichnis lost+found in der Wurzel des geprüften Dateisystems bereits existiert und freie Plätze für neue Einträge enthält. Um dies zu gewährleisten, legt man lost+found an, indem man einige Dateien in das Verzeichnis kopiert und diese dann wieder löscht (bevor fsck ausgeführt wird).

Das Prüfen des Raw-Geräts ist fast immer schneller.

Anhang 1: Administrator-Kommandos

Mit **dfsck** können Prüfungen für zwei verschiedene Dateisysteme auf unterschiedlichen Geräten gleichzeitig vorgenommen werden.

Das dfsck-Programm untersagt dem Bediener mit zwei fsck-Programmen gleichzeitig zu kommunizieren. Um dies zu umgehen, wird bei jeder Nachricht der Dateisystemname mit ausgegeben. Bei der Beantwortung einer Frage muß der Bediener seiner Antwort jeweils eine 1 oder eine 2 voranstellen, um somit auszudrücken, daß sich seine Antwort auf das erste oder das zweite Dateisystem bezieht.

Das Root-Dateisystem darf **nicht** mit dfsck überprüft werden.

DATEIEN:

/etc/checklist Standardliste der zu prüfenden Dateisysteme.

/etc/checkall dfsck-Shelldatei.

SIEHE AUCH:

Administrator-Kommandos: checkall, clri, ncheck

Dateiformate: checklist, fs

Anhang 1: Administrator-Kommandos

fsdb – Dateisystem-Debugger

SYNTAX:

/etc/fsdb Gerätedatei [-]

BESCHREIBUNG:

Mit fsdb kann ein beschädigtes Dateisystem nach einem Absturz wieder korrigiert werden. Das Kommando besitzt Möglichkeiten für die Umrechnung von Block- und I-Nummern in die entsprechenden Plattenadressen. Ferner sind mnemonische Offsets vorhanden, mit denen auf verschiedene Teile eines I-Knoten zugegriffen werden kann. Dadurch wird die Korrektur der Steuerblockeinträge und das Absteigen im Dateisystembaum wesentlich erleichtert.

Fsdb verfügt über mehrere Fehlerprüfroutinen, mit denen I-Knoten- und Blockadressen geprüft werden können. Diese können nötigenfalls deaktiviert werden, indem man fsdb mit dem optionalen Argument - aufruft oder das Symbol O verwendet. (Fsdb liest die Einträge i-size und f-size aus dem Superblock des Dateisystems und verwendet diese als Grundlage für die Prüfungen.)

Zahlen werden standardmäßig als dezimal angenommen. Oktalzahlen muß eine Null vorangestellt werden. Bei jeder Zuweisung werden Zahlen auf mögliche Abbruchfehler wegen einer ungleichen Länge von Quelle und Ziel geprüft.

Fsdb liest jeweils einen Block und arbeitet daher mit Raw E/A ebenso wie mit blockorientierter E/A. Eine Pufferverwaltungsroutine speichert häufig benutzte Datenblöcke, um die Anzahl der read-Systemaufrufe zu reduzieren. Bei einer Zuweisung wird der entsprechende Block sofort direkt geschrieben.

Anhang 1: Administrator-Kommandos

Fsdb erkennt die folgenden Symbole:

- # Absolute Adresse
- i Konvertieren von I-Nummer in I-Knoten-Adresse
- b Konvertieren in Blockadresse
- d Verzeichnisplatz-Offset
- +,- Adreßarithmetik
- q Quit (beenden)
- >,< Adresse speichern, Adresse einfügen
- = Numerische Zuweisung
- =+ Inkrementelle Zuweisung
- =- Dekrementelle Zuweisung
- = " Zeichenkettenzuweisung
- O Flip-Flop für Fehlerprüfung
- p Allgemeine Ausgabefunktionen
- f Dateiausgabefunktion
- B Byte-Modus
- W Wort-Modus
- D Doppelwort-Modus
- ! Aufrufen der Shell

Anhang 1: Administrator-Kommandos

Die Ausgabefunktionen erzeugen eine formatierte Ausgabe in unterschiedlichen Formen. Die aktuelle Adresse wird vor dem Ausgeben auf eine geeignete Grenze eingestellt. Sie wird während der Ausgabe hochgezählt und bleibt auf der Adresse des letzten ausgegebenen Elements stehen. Die Ausgabe kann jederzeit durch Eingabe des Delete-Zeichens beendet werden. Folgt auf das Symbol p eine Zahl, werden entsprechend viele Einträge ausgegeben. Es wird auf überlaufende Blockgrenzen geprüft, da logisch sequentielle Blöcke im allgemeinen nicht physikalisch sequentiell sind. Wird ein Zählerwert von Null angegeben, werden alle Einträge bis zum Ende des aktuellen Blocks ausgegeben. Folgende Ausgabeoptionen stehen zur Verfügung:

- i Als I-Knoten ausgeben
- d Als Verzeichnisse ausgeben
- o Als Oktalworte ausgeben
- e Als Dezimalworte ausgeben
- c Als Zeichen ausgeben
- b Als Oktalbytes ausgeben

Das Symbol f dient dazu, zum aktuellen I-Knoten gehörende Datenblöcke auszugeben. Schließt sich eine Zahl an, wird der betreffende Block der Datei ausgegeben. (Blöcke werden von Null aufwärts nummeriert.) Der Kennbuchstabe für die gewünschte Ausgabeoption steht hinter der Blocknummer, falls vorhanden, bzw. hinter dem Symbol f. Diese Ausgabefunktion ist auf kleine – ebenso wie auf große – Dateien anwendbar. Dabei wird auf spezielle Geräte geprüft, und es wird geprüft, daß die Blockzeiger, mit denen die Daten angesteuert werden, nicht Null sind.

Anhang 1: Administrator-Kommandos

Als Funktionsbegrenzungszeichen können Punkte, Tabs oder Leerzeichen verwendet werden, obwohl dies nicht unbedingt notwendig ist. Eine Zeile, die nur ein Neue-Zeile-Zeichen enthält, bewirkt, daß die aktuelle Adresse um die Größe des zuletzt ausgegebenen Datentyps erhöht wird. Das heißt, daß die Adresse auf das nächste Byte, Wort, Doppelwort, den nächsten Verzeichniseintrag oder I-Knoten gesetzt wird, so daß der Benutzer schrittweise einen Bereich des Dateisystems durchlaufen kann. Die Informationen werden in einem für den Datentyp geeigneten Format ausgegeben. Bei Bytes, Worten und Doppelworten wird die oktale Adresse und anschließend der Wert oktal und dezimal ausgegeben. An die Adresse für Byte- und Doppelwortwerte wird .B bzw. .D angehängt. Bei Verzeichnissen wird der Verzeichnisplatz-Offset und anschließend die dezimale I-Nummer und der Eintragsname im Zeichenformat ausgegeben. I-Knoten werden in Form von Feldern mit Feldbezeichnungen ausgegeben, die die einzelnen Elemente beschreiben.

Für die Untersuchung eines I-Knotens werden die folgenden mnemonischen Codes verwendet, die sich auf den aktuellen I-Knoten beziehen:

md	Modus
ln	Link-Anzahl
uid	Benutzer-ID-Nummer
gid	Gruppen-ID-Nummer
sz	Dateigröße
a#	Datenblocknummern (0–12)
at	Zugriffszeit
mt	Änderungszeit
maj	Major-Nummer
min	Minor-Nummer

Anhang 1: Administrator-Kommandos

BEISPIELE:

- 386i Gibt die I-Nummer 386 im I-Knoten-Format aus. Dieser wird dadurch der aktuelle I-Knoten.
- ln=4 Stellt den Link-Zähler für den aktuellen I-Knoten auf 4 ein.
- ln==1 Erhöht den Link-Zähler um 1.
- fc Gibt Block Null der zu dem aktuellen I-Knoten gehörenden Datei im ASCII-Format aus.
- 2i.fd Gibt die ersten 32 Verzeichniseinträge für den Wurzel-I-Knoten dieses Dateisystems aus.
- d5i.fc Macht den I-Knoten, der zu dem 5. Verzeichniseintrag (ab Null numeriert) gehört, der aufgrund des o. g. Kommandos gefunden wurde, zum aktuellen I-Knoten. Der erste logische Block der Datei wird danach im ASCII-Format ausgegeben.
- 512B.p0o Gibt den Superblock dieses Dateisystems oktal aus.
- 2i.a0b.d7=3 Ändert die I-Nummer für den siebten Verzeichnisplatz im Wurzelverzeichnis in 3 ab. Dieses Beispiel zeigt auch, wie mehrere Operationen in einer Kommandozeile zusammengefaßt werden können.
- d7.nm="Name" Ändert das Namensfeld auf dem Verzeichnisplatz in die angegebene Zeichenkette ab. Die Anführungszeichen können weggelassen werden, wenn das erste Zeichen ein Buchstabe ist.
- a2b.p0d Gibt den dritten Block des aktuellen I-Knoten als Verzeichniseinträge aus.

SIEHE AUCH:

Administrator-Kommando: fsck
Dateiformate: dir, fs

Anhang 1: Administrator-Kommandos

fuser – Prozesse identifizieren, die eine Datei oder Dateistruktur benutzen

SYNTAX:

```
/etc/fuser [-ku] Dateien [-] [[-ku] Dateien]
```

BESCHREIBUNG:

Fuser listet die Prozeß-IDs der Prozesse, die die als Argumente angegebenen Dateien benutzen, auf. Bei blockorientierten Gerädateien werden alle Prozesse, die eine beliebige Datei auf dem Gerät benutzen, aufgelistet. An die Prozeß-ID schließt sich c, p oder r an, wenn der Prozeß die Datei als aktuelles Verzeichnis, als Vater seines aktuellen Verzeichnisses (nur wenn vom System belegt) bzw. als sein Wurzelverzeichnis benutzt. Bei Angabe der Option -u schließt sich an die Prozeß-ID außerdem der Login-Name in Klammern an. Ist die Option -k angegeben ist, wird außerdem das Signal SIGKILL an jeden Prozeß abgesetzt. Nur der Superuser kann einen Prozeß eines anderen Benutzers terminieren (siehe Systemschnittstelle kill). Sie können verschiedene Optionen für verschiedene Dateien angeben. Die neuen Optionen ersetzen die alten. Ein - alleine setzt alle Optionen außer Kraft.

Die Prozeß-IDs werden in einer Zeile, getrennt durch Leerzeichen, auf der Standardausgabe ausgegeben. Am Ende steht ein einzelnes Zeilenendezeichen. Alle anderen Ausgaben werden in die Standardfehlerausgabe geschrieben.

BEISPIELE:

```
fuser -ku /dev/nd01
```

Dieses Kommando beendet alle Prozesse, die das Aushängen von Plattenlaufwerk 1 verhindern. Dieses Kommando muß vom Superuser aufgerufen werden. Die Prozeß-ID und der Login-Name jedes gekillten Prozesses werden ausgegeben.

Anhang 1: Administrator-Kommandos

```
fuser -u /etc/passwd
```

Dieses Kommando listet die Prozeß-IDs und Login-Namen der Prozesse, die die Paßwortdatei eröffnet haben.

```
fuser -ku /dev/nd01 -u /etc/passwd
```

Dieses Kommando erfüllt die gleichen Aufgaben, wie die beiden vorher beschriebenen Kommandos.

DATEIEN:

/unix	System-Namensliste
/dev/kmem	Speicherabbild (System Image)
/dev/mem	Speicherabbild (System Image)

SIEHE AUCH:

Administrator-Kommando: mount
Benutzerkommando: ps
Systemschnittstellen: kill, signal

Anhang 1: Administrator-Kommandos

fwtmp, wtmpfix – Bearbeitung von Anschluß-Abrechnungssätzen

SYNTAX:

```
/usr/lib/acct/fwtmp [-ic]  
/usr/lib/acct/wtmpfix [Dateien]
```

BESCHREIBUNG:

Fwtmp liest aus der Standardeingabe und schreibt in die Standardausgabe. Es konvertiert binäre Datensätze, wie sie in wtmp vorliegen, in formatierte ASCII-Sätze. Die ASCII-Version ermöglicht die Editierung fehlerhafter Sätze mit Hilfe von Editoren sowie die allgemeine Pflege der Datei.

Durch das Argument `-ic` können Sie angeben, daß die Eingabe ASCII-Format hat und die Ausgabe im Binärformat erfolgen soll.

Wtmpfix untersucht die Standardeingabe oder die angegebenen Dateien (wtmp-Format), korrigiert die Zeit-/Datumsangaben, um die Einträge konsistent zu machen, und gibt auf der Standardausgabe aus. Geben Sie als Dateinamen ein `- an`, bedeutet dies die Standardeingabe. Sofern Zeit- und Datumsangaben nicht korrigiert werden, endet `acctcon1` negativ, wenn es bestimmte Datumswechselsätze findet.

Jedesmal wenn das Datum gesetzt wird, wird ein Paar von Datumswechselsätzen in `/etc/wtmp` geschrieben. Der erste Satz ist das alte Datum, das durch die Zeichenkette `old time` und den Flag `OLD_TIME` im Type-Feld der Struktur `<utmp.h>` bezeichnet wird. Der zweite Satz gibt das neue Datum an und wird durch die Zeichenkette `new time` und den Flag `NEW_TIME` im Type-Feld bezeichnet. `Wtmpfix` greift auf diese Sätze zurück, um alle Zeitangaben in der Datei zu synchronisieren.

Anhang 1: Administrator-Kommandos

Neben der Korrektur der Zeit-/Datumsangaben überprüft wtmpfix auch die Gültigkeit des Namenfeldes, damit dieses ausschließlich alphanumerische Zeichen oder Leerzeichen enthält. Wenn es einen ungültigen Namen findet, ändert es den Login-Namen in INVALID ab und schreibt eine Diagnosemeldung in die Standardfehlerausgabe. So sorgt wtmpfix dafür, daß acctcon1 bei der Verarbeitung von Anschluß-Abrechnungssätzen seltener auf Fehler stößt.

DATEIEN:

/etc/wtmp

/usr/include/utmp.h

SIEHE AUCH:

Administrator-Kommandos: acct, acctcms, acctcon, acctmerg, acctprc, acctsh, runacct

Dateiformate: acct, utmp

Benutzerkommandos: acctcom, ed

Systemschnittstelle: acct

Anhang 1: Administrator-Kommandos

getty – Terminaltyp, Modi, Geschwindigkeit und Leitungsprozedur einstellen

SYNTAX:

```
/etc/getty [-h] [-l] [-t Timeout] Leitung [Geschwindigkeit  
[Typ [Leistungsprotokol]]]  
  
/etc/getty -c Datei
```

BESCHREIBUNG:

Getty ist ein Programm, das von init aufgerufen wird. Es ist der zweite Prozeß in der Serie der Prozesse (init → getty → login → shell), die einen Benutzer mit dem UNIX-System verbinden. Zunächst erzeugt getty eine Systemidentifikationsmeldung aus den Werten, die vom Systemaufruf `uname` zurückgeliefert werden. Ist die Datei `/etc/issue` vorhanden, gibt es diese Datei sodann am Benutzerterminal aus, und darauf folgt schließlich das Login-Meldungsfeld für den Eintrag aus `/etc/gettydefs`, den es benutzt. Getty liest den Login-Namen des Benutzers und ruft das Kommando `login` mit dem Benutzernamen als Argument auf. Während es den Namen liest, versucht getty, das System an die Geschwindigkeit und den Typ des jeweiligen Terminals anzupassen.

Leitung ist der Name einer Terminalleitung in `/dev`, die zum Lesen und Schreiben eröffnet werden soll. Wird getty nicht mit der Option `-h` aufgerufen, erzwingt es eine Unterbrechung der Leitung, indem es die Geschwindigkeit auf Null setzt, bevor es sie auf die standardmäßige oder angegebene Geschwindigkeit setzt. Durch die Option `-t` und die Angabe von *Timeout* in Sekunden wird festgelegt, daß getty terminiert, wenn die Eröffnung der Leitung erfolgreich verlaufen ist und innerhalb der angegebenen Anzahl Sekunden nichts eingegeben wird. Bei Angabe der Option `-l` erwartet getty erst das Laden eines Platzprogramms (DAP4x), bevor ein `login` gestartet wird.

Das optionale zweite Argument – *Geschwindigkeit* – ist eine Geschwindigkeitsangabe und eine Terminaldefinition in der Datei `/etc/gettydefs`. Aufgrund dieser Definition erkennt getty, welche Geschwindigkeit zunächst gefahren werden soll, wie die Login-Meldung auszusehen hat, welche Terminaleinstellungen anfangs gelten und welche Geschwindigkeit als nächstes benutzt werden soll, wenn der Be-

Anhang 1: Administrator-Kommandos

nutzer bekanntgibt, daß die Geschwindigkeit ungeeignet ist (<break>-Zeichen). Die standardmäßige Geschwindigkeit beträgt 300 Baud.

Das optionale dritte Argument, – *Typ* – ist eine Zeichenkette, aufgrund der *getty* erkennt, welcher Terminaltyp an die fragliche Leitung angeschlossen ist. *Getty* erkennt die folgenden Typen:

none	Standard
vt61	DEC vt61
vt100	DEC vt100
hp45	Hewlett-Packard HP45
c100	Concept 100

Der Standard-Terminaltyp ist *none* (keins), d. h. ein beliebiges dem System unbekanntes Terminal mit oder ohne Bildschirm. Damit der Terminaltyp überhaupt sinnvoll ausgewertet werden kann, müssen die virtuellen Terminal-Handler im Betriebssystem kompiliert sein. Sie sind verfügbar, sind aber in der Standardkonfiguration nicht kompiliert.

Das optionale vierte Argument – *Leitungsprotokoll* – ist eine Zeichenkette, die das Leitungsprotokoll für die Kommunikation mit dem Terminal beschreibt. Auch hier sind die Anschlußpunkte für die Leitungsprotokolle im Betriebssystem vorhanden, aber derzeit ist nur ein Leitungsprotokoll realisiert, und zwar das Standardprotokoll LDISC0.

Sind keine optionalen Argumente angegeben, setzt *getty* die Geschwindigkeit auf 300 Baud und legt folgendes fest:

- Raw-Mode (Aktivierung bei jedem Zeichen),
- kein echo,
- jede Parität ist erlaubt,
- Zeilenende-Zeichen werden in <CR> konvertiert,
- Tab-Expansion in der Standardausgabe.

© ... Weitergabe sowie Vervielfältigung dieser Unterlagen, Verwertung und Mitteilung ihres Inhalts nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zu Schadenersatz. Alle Rechte für den Fall der Patenterteilung oder Gebrauchsmustrentragung vorbehalten. ...

A1

Anhang 1: Administrator-Kommandos

Die Login-Meldung wird ausgegeben, bevor der Name des Benutzers zeichenweise gelesen wird. Trifft ein leeres Zeichen (oder ein Framing-Fehler) ein, wird dies als Betätigung der Break-Taste durch den Benutzer interpretiert. Daraufhin nimmt `getty` die nächste Geschwindigkeit in der Serie. Die Reihenfolge der von `getty` eingestellten Geschwindigkeiten hängt von dem Inhalt von `/etc/gettydefs` ab.

Der Benutzername wird durch ein Zeilenende-Zeichen oder `<CR>` beendet. Der Benutzername wird daraufhin untersucht, ob er Kleinbuchstaben enthält. Ist dies nicht der Fall und ist der Name nicht leer, so veranlaßt dies das System, alle künftigen Großbuchstaben in die entsprechenden Kleinbuchstaben umzusetzen.

Schließlich wird `login` mit dem Benutzernamen als Argument aufgerufen. Hinter dem Login-Namen können weitere Argumente eingegeben werden. Diese werden an `login` übergeben, das sie in die Umgebung einbringt.

Wird `getty` mit der Option `-c` und *Datei* aufgerufen, sucht es die angegebene Datei in derselben Weise wie `/etc/gettydefs` ab und gibt die Ergebnisse auf der Standardausgabe aus. Liegen nicht erkannte Modi oder fehlerhaft aufgebaute Einträge vor, so werden diese gemeldet. Sind die Einträge einwandfrei, gibt das Kommando die Werte der verschiedenen Eintragungen aus. Einige Werte werden den Eintragungen automatisch hinzugefügt.

DATEIEN:

`/etc/gettydefs`

`/etc/issue`

SIEHE AUCH:

Administrator-Kommando: `init`

Dateiformate: `gettydefs`, `inittab`

Benutzerkommando: `login`

Gerätedatei: `tty`

Systemschnittstelle: `ioctl`

Anhang 1: Administrator-Kommandos

grpck, pwck – Prüfen der Gruppen- bzw. Paßwortdatei

SYNTAX:

```
/etc/grpck [Date]
/etc/pwck [Date]
```

BESCHREIBUNG:

Pwck überprüft die Paßwortdatei und meldet gegebenenfalls gefundene Inkonsistenzen. Das Kommando überprüft die Anzahl der Felder, den Login-Namen, die Benutzer-ID, die Gruppen-ID und ob das Login-Verzeichnis und der wahlfreie Programmname existieren. Geben Sie keine andere Datei an, wird /etc/passwd überprüft.

Grpck überprüft alle Einträge in der Gruppdatei. Es prüft die Anzahl der Felder, den Gruppennamen, die Gruppen-ID und ob alle Login-Namen auch in der Paßwortdatei eingetragen sind. Geben Sie keine Datei an, wird /etc/group überprüft.

DATEIEN:

```
/etc/group
/etc/passwd
```

SIEHE AUCH:

Dateiformate: group, passwd

Anhang 1: Administrator-Kommandos

init, telinit – Initialisierung der Prozeßsteuerung

SYNTAX:

```
/etc/init [0123456SsQq]  
/etc/telinit [0123456sSQqabc]
```

BESCHREIBUNG:

Init dient zum Erzeugen von Prozessen. Es hat in erster Linie die Aufgabe, Prozesse aufgrund eines in der Datei `/etc/inittab` abgelegten Scripts zu erzeugen. Diese Datei veranlaßt `init` in der Regel, `getty`s auf jeder Leitung zu erzeugen, auf der sich ein Benutzer anmelden kann. Ferner steuert das Kommando autonome Prozesse, die in einem bestimmten System benötigt werden.

`Init` geht davon aus, daß das System sich zu jedem gegebenen Zeitpunkt in irgendeinem Run-Level (Arbeitsumgebung) befindet. Ein Run-Level kann als Softwarekonfiguration des Systems angesehen werden, wobei in jeder Konfiguration nur eine ausgewählte Gruppe von Prozessen existieren kann. Die von `init` für jeden dieser Run-Levels erzeugten Prozesse sind in der Datei `inittab` definiert. `Init` kann sich jeweils in einem von acht Run-Levels befinden, und zwar 0–6 und S oder s. Der Run-Level wird geändert, wenn ein privilegierter Benutzer `/etc/init` ausführt (verknüpft mit `/etc/telinit`). Dieses vom Benutzer erzeugte `init` sendet entsprechende Signale an das ursprüngliche vom Betriebssystem beim Neustart des Systems erzeugte `init` und teilt ihm mit, in welchen Run-Level gewechselt werden soll.

`Init` wird im UNIX-System als letzter Schritt beim Systemstart aufgerufen. Als erstes sucht `init` die Datei `/etc/inittab` und prüft, ob diese einen Eintrag des Typs `initdefault` enthält. Wenn ja, tritt `init` zunächst in den in diesem Eintrag angegebenen Run-Level ein. Befindet sich dieser Eintrag nicht in `inittab` oder wird `inittab` nicht gefunden, so verlangt `init` vom Benutzer, daß er einen Run-Level an der virtuellen Systemkonsole `/dev/syscon` eingeben möge. Wird ein S (s) eingegeben, geht `init` in den SINGLE USER-Level. Dies ist der einzige Run-Level, der nicht das Vorhandensein einer sachgemäß formatierten `inittab`-Datei voraussetzt. Wenn `/etc/inittab` nicht existiert, ist der SINGLE USER-Modus standardmäßig der einzige zulässige Run-Level, in den `init` übergehen kann. Im

Anhang 1: Administrator-Kommandos

SINGLE USER-Modus wird die virtuelle Konsole /dev/syscon zum Lesen und Schreiben eröffnet und das Kommando /bin/su sofort aufgerufen. Zum Verlassen des SINGLE USER-Modus gibt es zwei Möglichkeiten:

- Die Shell wird durch ein Dateieinde-Zeichen terminiert; init verlangt dann die Eingabe eines neuen Run-Levels.
- Init oder telinit können über Signale eine Änderung des Run-Levels erzwingen.

Fordert init beim Neustarten des Systems keinen neuen Run-Level an, kann dies darauf zurückzuführen sein, daß das Gerät /dev/syscon mit einem anderen Gerät als der physischen Systemkonsole (/dev/systty) verknüpft ist. In diesem Fall kann init veranlaßt werden, /dev/syscon neu zu verknüpfen, wenn Sie ein delete auf der Systemkonsole eingeben, die dem Prozessor zugeordnet ist.

Verlangt init die Eingabe eines neuen Run-Levels, dürfen Sie nur eine der Ziffern von 0 bis 6 oder einen der Buchstaben S oder s eingeben. Wird S eingegeben, arbeitet init so wie oben beschrieben im SINGLE USER-Modus, wobei aber zusätzlich /dev/syscon mit der Terminalleitung des Benutzers verknüpft wird, so daß diese zur virtuellen Systemkonsole wird. Auf der physischen Konsole /dev/systty wird eine Meldung erzeugt, die angibt, wohin das virtuelle Terminal gelegt wurde.

Beim Systemstart und immer wenn init vom SINGLE USER-Modus in normale Run-Level umschaltet, setzt es die ioctl-Statik der virtuellen Konsole /dev/syscon auf die Modi, die in der Datei /etc/ioctl.syscon gespeichert sind. Diese Datei wird von init immer dann beschrieben, wenn das System in den SINGLE USER-Modus übergeht. Ist diese Datei nicht vorhanden, wenn init sie lesen möchte, wird eine Warnung ausgegeben und Standard-Einstellungen vorgenommen.

Wird eine der Ziffern 0 bis 6 eingegeben, tritt init in den entsprechenden Run-Level ein. Jede andere Eingabe wird zurückgewiesen, und der Benutzer wird erneut zur Eingabe aufgefordert. Ist dies das erste Mal, daß init in einen anderen Run-Level als S oder s überwechselt, durchsucht init zunächst inittab nach speziellen Einträgen des Typs boot und bootwait. Diese Einträge werden ausgeführt, sofern der aufgerufene Run-Level mit dem des Eintrags übereinstimmt, bevor die normale

Anhang 1: Administrator-Kommandos

Verarbeitung von `inittab` erfolgt. So ist es möglich, daß gegebenenfalls erforderliche besondere Initialisierungen des Betriebssystems, z. B. das Einhängen von Dateisystemen, vorgenommen werden können, bevor die Benutzer mit dem System arbeiten können. Die `inittab`-Datei wird nach allen Einträgen durchsucht, die für den betreffenden Run-Level zu verarbeiten sind.

Run-Level 2 ist normalerweise so definiert, daß er alle Terminalprozesse und Dämonen enthält, die in der Multi-User-Umgebung erzeugt werden.

In einer Multi-User-Umgebung wird die `inittab`-Datei in der Regel so eingerichtet, daß `init` einen Prozeß für jedes Terminal im System erzeugt.

Bei Terminalprozessen endet die Shell letzten Endes aufgrund eines Dateiende-Zeichens, das entweder explizit eingegeben oder durch eine Leitungsunterbrechung erzeugt wird. Wenn `init` ein Child-Death-Signal empfängt, welches besagt, daß ein von `init` erzeugter Prozeß nicht mehr existiert, verzeichnet es diesen Umstand und den Grund der Beendigung des Prozesses in `/etc/utmp` und `/etc/wtmp`, falls vorhanden. Ein Protokoll der erzeugten Prozesse wird in `/etc/wtmp` geführt, falls diese Datei existiert.

Um die einzelnen Prozesse in der `inittab`-Datei zu erzeugen, liest `init` die einzelnen Einträge, und bei jedem Eintrag, der neu erzeugt werden soll, bringt es einen Kindprozeß hervor. Nachdem `init` alle in der `inittab`-Datei angegebenen Prozesse erzeugt hat, wartet es, bis einer der Kindprozesse terminiert, ein Netzausfall-Signal eintrifft oder `init` von `init` oder `telinit` per Signal aufgefordert wird, den Run-Level des Systems zu ändern. Tritt eine der genannten drei Bedingungen ein, sieht `init` erneut die `inittab`-Datei durch. Es können jederzeit neue Einträge in die `inittab`-Datei eingebracht werden; allerdings wartet `init` in jedem Fall darauf, daß eine der drei oben genannten Bedingungen eintritt. Um eine sofortige Reaktion zu ermöglichen, kann `init` mit dem Kommando `init Q` oder `init q` veranlaßt werden, die `inittab`-Datei erneut zu durchsuchen.

Anhang 1: Administrator-Kommandos

Empfängt init ein Netzausfallsignal (SIGPWR) und befindet sich nicht im SINGLE USER-Modus, sucht es inittab nach speziellen Netzausfall-einträgen ab. Diese Einträge werden vor jeder weiteren Verarbeitung aufgerufen (sofern die Run-Levels dies erlauben). Auf diese Weise kann init immer dann, wenn das Betriebssystem einen Netzausfall feststellt, verschiedene Bereinigungs- und Protokollierungsfunktionen wahrnehmen.

Wird init (über telinit) aufgefordert, den Run-Level zu wechseln, sendet init das Warnsignal (SIGTERM) an alle Prozesse, die im neuen Run-Level nicht definiert sind. Init wartet 20 Sekunden, bevor es diese Prozesse mit dem Kill-Signal (SIGKILL) zwangsweise beendet.

Mit **telinit**, das mit /etc/init verknüpft ist, werden die Aktionen von init gesteuert. Es wird mit einem einzelnen Argument aufgerufen und veranlaßt init über die Systemschnittstelle kill, die entsprechenden Aktionen auszuführen. Folgende Argumente wirken als Anweisungen auf init:

- 0-6 Veranlaßt init, daß System in einen der Run-Level 0-6 zu setzen.
- a,b,c Veranlaßt init, nur die Einträge in der Datei /etc/inittab zu bearbeiten, in denen der Run-Level a, b oder c gesetzt ist.
- Q,q Veranlaßt init, die Datei etc/inittab zu durchsuchen.
- s,S Veranlaßt init, in den Single-User-Modus umzuschalten. Wird dieser Run-Level-Wechsel durchgeführt, wird die virtuelle Systemkonsole /dev/syscon das Terminal, von dem das Kommando abgesetzt wurde.

Telinit kann nur vom Superuser oder einem Mitglied der Gruppe sys ausgeführt werden.

Anhang 1: Administrator-Kommandos

DATEIEN:

/etc/inittab
/etc/utmp
/etc/wtmp
/etc/loctl.syscon
/dev/syscon
/dev/systty

MELDUNGEN:

Stellt init fest, daß es für einen Eintrag in /etc/inittab öfter als zehn Mal innerhalb von zwei Minuten Prozesse erzeugt, geht es davon aus, daß im Kommando-String ein Fehler vorliegt, gibt eine Fehlermeldung auf der Systemkonsole aus und verweigert für die Dauer von fünf Minuten oder bis es ein Signal von einem Benutzer-init (telinit) empfängt, die Erzeugung neuer Prozesse für diesen Eintrag. Dadurch wird verhindert, daß init Systemressourcen belegt, wenn die inittab-Datei fehlerhaft editiert ist oder bereits gelöschte Programme angesprochen werden.

SIEHE AUCH:

Administrator-Kommando: getty
Dateiformate: inittab, utmp
Benutzerkommandos: login, sh, who
Systemschnittstelle: kill

Anhang 1: Administrator-Kommandos

install – Installationskommandos

SYNTAX:

/etc/install [-c Verzeichnis A] [-f Verzeichnis B] [-i] [-n Verzeichnis C] [-o] [-s] Datei [Verzeichnis X ...]

BESCHREIBUNG:

Das Kommando `install` wird meist in Makedateien verwendet, um eine Datei (aktualisierte Zieldatei) an einer bestimmten Stelle innerhalb eines Dateisystems zu installieren. Eine Datei wird installiert, indem man sie in das entsprechende Verzeichnis kopiert und dadurch die Zugriffsrechte und den Eigentümer des ursprünglichen Kommandos beibehält. Das Programm gibt Meldungen aus, die dem Benutzer im einzelnen Aufschluß darüber geben, welche Dateien ersetzt bzw. erstellt werden und an welche Stelle sie gebracht werden.

Sind keine Optionen oder Verzeichnisse angegeben, durchsucht `search` eine Reihe von Standardverzeichnissen (`/bin`, `/usr/bin`, `/etc`, `/lib` und `/usr/lib` in dieser Reihenfolge) nach einer Datei mit demselben Namen den Sie als Dateinamen beim Kommandoaufruf angegeben haben. Sobald dieser Dateiname gefunden wird, teilt `install` durch eine Meldung mit, daß es die gefundene Datei mit der angegebenen Datei überschreibt und tut dies. Wird keine im System vorhandene Datei mit dem angegebenen Namen gefunden, wird dies ebenfalls durch eine Nachricht mitgeteilt und das Programm ist beendet.

Geben Sie hinter `Datei` ein oder mehrere Verzeichnisse an, werden diese Verzeichnisse vor den Verzeichnissen der Standardliste durchsucht.

Optionen:

`-c Verzeichnis A`

Installiert `Datei` in dem von Ihnen angegebenen Verzeichnis nur dann, wenn die Datei nicht gefunden wird. Wird sie gefunden, teilt `install` durch eine Meldung mit, daß diese bereits existiert, und das Kommando terminiert, ohne daß die Datei überschrieben wird. Diese Option kann allein oder zusammen mit der Option `-s` verwendet werden.

Anhang 1: Administrator-Kommandos

-f *Verzeichnis B*

Datei wird in jedem Fall in dem angegebenen Verzeichnis installiert, auch wenn sie bereits existiert. Wenn die zu installierende Datei noch nicht existiert, werden die Zugriffsrechte und der Eigentümer der neuen Datei auf 755 bzw. bin gesetzt. Existiert die Datei bereits, werden die Zugriffsrechte und der Eigentümer der alten Datei beibehalten. Diese Option kann allein oder zusammen mit den Optionen -o oder -s verwendet werden.

- i Die Standard-Verzeichnisliste wird ignoriert, und es werden nur die angegebenen Verzeichnisse (*Verzeichnis X ...*) durchsucht. Diese Option kann allein oder zusammen mit allen anderen Optionen außer -c und -f angegeben werden.

-n *Verzeichnis C*

Wird die angegebene Datei in keinem der durchsuchten Verzeichnisse gefunden, wird sie in dem mit dieser Option angegebenen Verzeichnis installiert. Die Zugriffsrechte und der Eigentümer der neuen Datei werden auf 755 bzw. bin gesetzt. Diese Option kann allein oder zusammen mit allen anderen Optionen außer -c und -f angegeben werden.

- o Ist die angegebene Datei bereits im System vorhanden, wird die gefundene Datei gerettet, indem sie in *OLDdatei* in dem Verzeichnis, in dem sie gefunden wurde, kopiert wird. Diese Option ist sinnvoll, wenn eine sehr aktive Datei wie /bin/sh oder /etc/getty installiert wird und deshalb die vorhandene Datei nicht gelöscht werden kann. Diese Option kann allein oder zusammen mit allen anderen Optionen außer -c angegeben werden.
- s Unterdrückt die Ausgabe von Meldungen außer Fehlermeldungen. Diese Option kann allein oder zusammen mit allen anderen Optionen verwendet werden.

SIEHE AUCH:

Benutzerkommando: make
Systemliteratur TARGON: „Programmentwicklungs-Tools“

Anhang 1: Administrator-Kommandos

killall – Alle aktiven Prozesse abbrechen

SYNTAX:

`/etc/killall [Signal]`

BESCHREIBUNG:

Killall ist eine Prozedur, mit deren Hilfe `/etc/shutdown` alle aktiven Prozesse abbricht, die nicht direkt mit der `shutdown`-Prozedur zusammenhängen.

Killall dient hauptsächlich dazu, alle Prozesse mit eröffneten Dateien zu beenden, so daß die eingehängten Dateisysteme deaktiviert werden und ausgehängt werden können.

Killall sendet das angegebene Signal an alle die Prozesse, die nicht zu der oben genannten Gruppe von Prozessen gehören. Geben Sie kein Signal, wird Signal 9 gesendet.

DATEIEN:

`/etc/shutdown`

SIEHE AUCH:

Administrator-Kommandos: `fuser`, `shutdown`

Benutzerkommandos: `kill`, `ps`

Systemschnittstelle: `signal`

Anhang 1: Administrator-Kommandos

link, unlink – Systemschnittstellen **link** und **unlink** aufrufen

SYNTAX:

/etc/link Datei1 Datei2

/etc/unlink Datei

BESCHREIBUNG:

Link und **unlink** führen die entsprechenden Systemaufrufe für die angegebenen Argumente ohne jegliche Fehlerprüfung aus.

Diese Kommandos können nur vom Superuser aufgerufen werden, der (hoffentlich) weiß, was er tut.

SIEHE AUCH:

Benutzerkommando: **rm**

Systemschnittstellen: **link**, **unlink**

Anhang 1: Administrator-Kommandos

lpadmin – Konfigurierung des Spool-Systems

SYNTAX:

```
/usr/lib/lpadmin -pDrucker [Optionen]
/usr/lib/lpadmin -xZiel
/usr/lib/lpadmin -d[Ziel]
```

BESCHREIBUNG:

Mit lpadmin werden Spool-Systeme konfiguriert, d. h. Drucker, Druckerklassen und Gerätedateien beschrieben. Mit diesem Kommando werden Ziele (Drucker oder Druckerklassen) eingerichtet und gelöscht, die Zugehörigkeit zu Klassen geändert, Gerätedateien für Drucker gewechselt, Schnittstellenprogramme für Drucker gewechselt und das Standardsystemziel geändert. Lpadmin kann außer in den unten ausdrücklich angegebenen Fällen nicht aufgerufen werden, wenn der Scheduler lpsched läuft.

Bei jedem Aufruf von lpadmin muß eine und nur eine der Optionen -p, -d oder -x angegeben werden.

Optionen:

- d[Ziel] *Ziel* muß ein bereits vorhandenes Ziel sein und wird hierdurch zum neuen Standardsystemziel. Geben Sie *Ziel* nicht an, so gibt es in dem System kein Standardziel. Diese Option kann verwendet werden, während der Scheduler lpsched läuft. Zusammen mit -d dürfen keine weiteren Optionen angegeben werden.
- x*Ziel* Entfernt das angegebene Ziel aus dem Spool-System. Ist *Ziel* ein Drucker und das einzige Mitglied einer Klasse, wird die Klasse ebenfalls gelöscht. Mit -x können keine weiteren Optionen angegeben werden.
- p*Drucker* Benennt einen Drucker, auf den die unten aufgeführten Optionen wirken. Wenn *Drucker* nicht existiert, wird er angelegt.

Anhang 1: Administrator-Kommandos

Die folgenden Optionen sind nur bei `-p` sinnvoll und können in beliebiger Reihenfolge angegeben werden. Der Einfachheit halber wird der Drucker im folgenden als `D` bezeichnet.

- `-cklasse` Bindet den Drucker `D` in die angegebene Klasse ein. Ist *Klasse* noch nicht vorhanden, wird sie angelegt.
- `-eDrucker` Kopiert das Schnittstellenprogramm des angegebenen Druckers, so daß dieses auch für `D` verwendet wird.
- `-h` Gibt an, daß das mit `D` verbundene Gerät fest verdrahtet ist. Diese Option wird beim Anlegen eines neuen Druckers standardmäßig angenommen, außer wenn die Option `-l` angegeben wird.
- `-iPfadname` Legt ein neues Schnittstellenprogramm für `D` an. Der angegebene Pfadname ist derjenige des neuen Programms.
- `-l` Zeigt an, daß das mit `D` verbundene Gerät ein Login-Terminal ist. Beim Start des Schedulers wird allen Terminals automatisch der Status `disabled` zugewiesen. Bevor `D` wieder mit `enable` aktiviert wird, sollte seine aktuelle Gerätedatei mit Hilfe von `lpadmin` eingerichtet werden.
- `-mModell` Auswahl einer Modell-Schnittstelle für `D`. *Modell* ist der Name einer der Modell-Schnittstellen, die im Lieferumfang der Software des Spool-Systems enthalten sind.
- `-rKlasse` Entfernt `D` aus der angegebenen Klasse. War `D` das letzte Mitglied der Klasse, wird auch die Klasse gelöscht.
- `-vGerät` Verbindet ein neues Gerät mit Drucker `D`. *Gerät* ist der Pfadname einer Datei, in die der Spooler-Administrator (Benutzername `lp`) schreiben darf. Der Administrator kann nicht daran gehindert werden, dieselbe Gerätedatei mit mehreren Druckern zu verwenden. Geben Sie nur die Optionen `-p` und `-v` an, können Sie `lpadmin` auch während der Laufzeit des Schedulers aufrufen.

Anhang 1: Administrator-Kommandos

Einschränkungen

Wird ein neuer Drucker angelegt, muß die Option -v und eine der Optionen -e, -i oder -m angegeben werden. Von den Optionen -e, -i und -m dürfen Sie nur eine einzige aufrufen. Die Optionen -h und -l schließen sich gegenseitig aus. Drucker- und Klassennamen können maximal 14 Stellen lang sein und dürfen nur die Zeichen A-Z, a-z, 0-9 und den Unterstrich (_) enthalten.

Modelle

In der Spooler-Software sind Modelle für Drucker-Schnittstellenprogramme enthalten. Dabei handelt es sich um Shell-Prozeduren, die eine Schnittstelle zwischen Ipsched und Geräten bilden. Alle Modelle sind im Verzeichnis /usr/spool/lp/model untergebracht und können mit lpadmin -m in unveränderter Form verwendet werden. Alternativ kann der Spooler-Administrator Kopien der Modelle modifizieren und diese über lpadmin -i mit Druckern verbinden. Im folgenden sind die Modelle beschrieben. Dabei sind jeweils die Optionen angegeben, mit denen sie in der lp-Kommandozeile, in der der Kennbuchstabe -o steht, aufgerufen werden können:

- dumb Schnittstelle für einen Zeilendrucker ohne Spezialfunktionen und Protokoll. Formularvorschübe werden implizit angenommen. Dieses Modell eignet sich gut zum Kopieren und Modifizieren für solche Drucker, für die keine Modelle vorhanden sind.
- 1640 DIABLO 1640 Terminal mit 1200 Baud und dem Protokoll XON/XOFF.
 - 12 12 Zeichen pro Zoll (10 Zeichen pro Zoll sind Standard).
 - f Den 450-Filter nicht verwenden. Die Ausgabe ist mit 450 oder nroff 450-Steuertabelle vorverarbeitet.
- hp Hewlett-Packard 2631A Zeilendrucker mit 2400 Baud.
 - c Komprimierter Druck.
 - e Expandierter Druck.
- prx Printronix P300 oder P600 Drucker mit XON/XOFF-Protokoll und 1200 Baud.

Anhang 1: Administrator-Kommandos

BEISPIELE:

1. Wenn wir davon ausgehen, daß ein Zeilendrucker Hewlett-Packard 2631A mit dem Namen hp2 vorhanden ist, benutzt dieser aufgrund des folgenden Kommandos die Modellschnittstelle hp:

```
/usr/lib/lpadmin -php2 -mhp
```

2. Mit dem folgenden Kommando erreichen Sie komprimiertes Drucken auf dem Drucker hp2:

```
lp -dhp2 -o-c Dateien
```

3. Mit folgendem Kommando können Sie einen Drucker DIABLO 1640 mit dem Namen st1 in die Spooler-Konfiguration aufnehmen:

```
/usr/lib/lpadmin -pst1 -v/dev/tty20 -m1640
```

4. Ein nroff-Dokument kann mit jedem der folgenden Kommandos auf st1 gedruckt werden:

```
nroff -T450 Dateien | lp -dst1 of  
nroff -T450- 12 Dateien | lp -dst1 -of  
nroff -T37 Dateien | col | lp -dst1
```

5. Das folgende Kommando druckt die Gruppendatei mit 12 Zeichen pro Zoll auf st1 aus:

```
lp -dst1 -o12 /etc/group
```

HINWEIS: Die Option -12 beim Drucker DIABLO 1640 darf nicht in Verbindung mit nroff verwendet werden.

DATEIEN:

```
/usr/spool/lp/*
```

SIEHE AUCH:

Administrator-Kommandos: accept, lpsched

Benutzerkommandos: enable, lp, lpstat, nroff

Anhang 1: Administrator-Kommandos

lpsched, lpshut, lpmove – Spooler-Scheduler starten/stoppen und Druckaufträge verschieben

SYNTAX:

```
/usr/lib/lpsched
/usr/lib/lpshut
/usr/lib/lpmove Aufträge Ziel
/usr/lib/lpmove Ziel1 Ziel2
```

BESCHREIBUNG:

lpsched steuert den zeitlichen Ablauf der von lp angenommen Druckaufträge.

lpshut stoppt den Spooler-Scheduler. Alle Drucker, die beim Aufruf von lpshut arbeiten, werden angehalten. Druckaufträge, die beim Stoppen eines Druckers gerade in Arbeit waren, werden nach dem erneuten Starten von lpsched vollständig neu gedruckt.

lpmove verschiebt von lp in die Warteschlange gestellte Druckaufträge von einem Ziel (Drucker oder Druckerklasse) zu einem anderen. Dieses Kommando kann nur benutzt werden, wenn der Scheduler nicht aktiv ist.

Bei der ersten Form des Kommandos werden die angegebenen Aufträge zu dem angegebenen Ziel (Drucker oder Druckerklasse) verschoben. Aufträge sind Auftrags-IDs, die von lp zurückgeliefert werden. Bei der zweiten Form werden alle Druckaufträge für *Ziel1* zum *Ziel2* verschoben. Als Nebeneffekt weist lp jetzt Druckaufträge für *Ziel1* zurück.

Beachten Sie, daß lpmove den Annahmestatus (siehe accept) für das neue Ziel beim Verschieben von Druckaufträgen nicht prüft.

Anhang 1: Administrator-Kommandos

DATEIEN:

/usr/spool/lp/*

SIEHE AUCH:

Administrator-Kommandos: accept, lpadmin

Benutzerkommandos: enable, lp, lpstat

Anhang 1: Administrator-Kommandos

mkfs – Dateisystem aufbauen

SYNTAX:

/etc/mkfs Gerätedatei Blöcke[:I-Knoten] [Blocklücke Blöcke/Zy/]

/etc/mkfs Gerätedatei Prototyp [Blocklücke Blöcke/Zy/]

BESCHREIBUNG:

Mkfs baut ein Dateisystem auf, indem es entsprechend den Anweisungen im weiteren Teil der Kommandozeile Werte in die angegebene Gerätedatei schreibt. Das Kommando wartet 10 Sekunden, bis es mit dem Aufbau des Dateisystems beginnt. Wenn das zweite Argument in Form einer Folge von Ziffern angegeben ist, baut mkfs ein Dateisystem mit einem einzelnen leeren Verzeichnis auf. Die Größe des Dateisystems ist die Angabe der Blöcke, die als Dezimalzahl interpretiert wird. Diese Angabe ist die Anzahl der physikalischen Plattenblöcke, die das Dateisystem belegt. Das Boot-Programm bleibt uninitialisiert. Wird die optionale Anzahl I-Knoten nicht angegeben, wird standardmäßig die durch vier dividierte Anzahl der logischen Blöcke genommen.

In der zweiten Form des Kommandoaufrufs wird das zweite Argument als Name einer Datei interpretiert. Kann diese Datei eröffnet werden, nimmt mkfs an, daß es sich um eine Prototypdatei handelt, und entnimmt seine Anweisungen aus dieser Datei. Die Prototypdatei enthält Tokens, die durch Leerzeichen oder Zeilenendezeichen voneinander getrennt sind. Das erste Token ist der Name einer Datei, die als Bootstrap-Programm in Block 0 kopiert werden soll. Das zweite Token ist eine Zahl, die die Größe des angelegten Dateisystems in physikalischen Plattenblöcken angibt. In der Regel wird dies die Anzahl Blöcke auf dem Gerät, möglicherweise vermindert um den für das Swapping benötigten Speicherplatz, sein. Das nächste Token ist die Anzahl der I-Knoten im Dateisystem. Es können maximal 65.500 I-Knoten vorgesehen werden. Die nächste Gruppe von Tokens enthält die Spezifikation für die Wurzeldatei. Dateispezifikationen bestehen aus Tokens, die den Zugriffsmodus, die Benutzer-ID, die Gruppen-ID und den anfänglichen Inhalt der Datei angeben. Die Syntax des Inhaltsfeldes hängt vom Zugriffsmodus ab.

Anhang 1: Administrator-Kommandos

Das Modus-Token für eine Datei ist eine sechsstellige Zeichenkette. Das erste Zeichen gibt den Typ der Datei an. (Die Zeichen -bcd stehen für Datendatei, blockorientierte Gerätedatei, zeichenorientierte Gerätedatei oder Verzeichnis.) Das zweite Zeichen ist entweder u oder – und bedeutet Set-User-ID-Bit gesetzt oder nicht. Das dritte Zeichen ist g oder – und steht für Set-Group-ID-Bit gesetzt oder nicht. Der Rest des Modus besteht aus einer dreistelligen Oktalzahl, die die Lese-, Schreib- und Ausführungserlaubnisse für den Eigentümer, die Gruppe und den „Rest der Welt“ angibt. Nach dem Modus kommen zwei Tokens in Form von Dezimalzahlen; sie geben die Benutzer-ID und die Gruppen-ID des Eigentümers der Datei an.

Handelt es sich bei der Datei um eine Datendatei, ist das nächste Token ein Pfadname, von dem der Inhalt und die Größe kopiert werden. Handelt es sich bei der Datei um eine blockorientierte oder zeichenorientierte Gerätedatei, schließen sich zwei dezimale Tokens an, die die Major- bzw. die Minor-Gerätenummer angeben. Handelt es sich bei der Datei um ein Verzeichnis, legt mkfs die Einträge . und .. an und liest sodann eine Liste mit Namen und (rekursiv) Dateispezifikationen für die Einträge im Verzeichnis. Der Lesevorgang wird mit dem Token \$ beendet.

Anhang 1: Administrator-Kommandos

Es folgt ein Muster für Prototyp-Spezifikationen:

```

/stand/diskboot
4872 110
d--777 3 1
usr  d--777 3 1
      sh  ---755 3 1 /bin/sh
      ken d--755 6 1
      $
      b0  b--644 3 1 0 0
      c0  c--644 3 1 0 0
      $
$
    
```

Bei beiden Syntaxformen des Kommandos kann die Blocklücke und die Anzahl der Blöcke/Zylinder angegeben werden. Die folgenden Werte werden empfohlen:

Gerät	Blocklücke	Blocks/Zyl
RL01/02	7	40
RP03	5	200
RP04/05/06	7	418
RP07	7	400
RM03	7	160
RM05	7	608
RM80	9	434
3B 20S computer MHD	7	608
default	7	400

Der Standardwert wird angenommen, wenn die angegebenen Werte für Blocklücke und Blöcke/Zylinder ungültig sind oder zu wenige Argumente vorhanden sind.

„Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und
 Mitteilung ihres Inhalts nicht gestattet, soweit nicht ausdrücklich zugestanden.
 Zuwiderhandlungen verpflichten zu Schadenersatz. Alle Rechte für den Fall
 der Patentierung oder Gebrauchsmusteranmeldung vorbehalten.“

A1

Anhang 1: Administrator-Kommandos

HINWEIS:

Wird ein Prototyp verwendet, kann eine Datei nur bis maximal 64 KByte initialisiert werden und gibt es keine Möglichkeit, Verknüpfungen (Links) zu spezifizieren.

SIEHE AUCH:

Benutzerkommando: `chmod`

Dateiformate: `dir`, `fs`

Anhang 1: Administrator-Kommandos

mknod – Anlegen einer Geräte- oder FIFO-Datei (Named-Pipe)

SYNTAX:

/etc/mknod Name b/c Major Minor

/etc/mknod Name p

BESCHREIBUNG:

Anlegen einer Gerätedatei

Dazu ist die erste Form des Aufrufs zu verwenden. *Name* ist der gewünschte Dateiname der Gerätedatei. Ist die Datei blockorientiert (Platten, Bänder), ist der zweite Parameter ein *b*. Bei zeichenorientierten Dateien ein *c*.

Major- und Minor-Gerätenummern sind systemspezifisch und können der entsprechenden Dokumentation oder der Systemdatei *conf.c* entnommen werden.

Anlegen einer Named-Pipe

Mit der zweiten Form des Aufrufs wird eine FIFO-Datei (Named-Pipe) angelegt.

SIEHE AUCH:

Systemschnittstelle: *mknod*

Anhang 1: Administrator-Kommandos

mount, umount – Dateisystem ein- und aushängen

SYNTAX:

```
/etc/mount [Name Verzeichnis[-r]]
```

```
/etc/umount Name
```

BESCHREIBUNG:

Name bezeichnet die Gerätedatei einer logischen Platte (Major-Nummer 107). *Verzeichnis* wird der Name des Root-Verzeichnisses des montierten Dateisystems. Ein Verzeichnis mit diesem Namen muß im Root-Dateisystem angelegt sein.

Soll das montierte Dateisystem nur lesbar sein, so ist die Option `-r` anzugeben. Die Dateisysteme können nur auf Platten betrieben werden, die nicht physikalisch schreibgeschützt sind.

Mount und **umount** verwalten eine Mount-Tabelle. Rufen Sie **mount** ohne Parameter auf, dann wird diese Tabelle ausgedruckt.

Umount demontiert das Dateisystem der logischen Platte *Name*.

HINWEIS:

Bevor Sie **mount** aufrufen, sollten Sie das entsprechende Dateisystem mit **fsck** überprüfen.

MELDUNGEN:

Mount gibt eine Warnung aus, wenn das angegebene Verzeichnis nicht mit dem Verzeichnis übereinstimmt, auf das das Dateisystem zuvor montiert war.

Umount wird nicht ausgeführt, wenn das Dateisystem aktiv ist. Ein Dateisystem ist aktiv, wenn eine Datei eröffnet ist oder ein Benutzer im Dateisystem arbeitet.

Anhang 1: Administrator-Kommandos

DATEIEN:

/etc/mnttab Mount-Tabelle

SIEHE AUCH:

Administrator-Kommandos: mkfs, fsck

Dateiformat: mnttab

Systemschnittstelle: mount

Anhang 1: Administrator-Kommandos

mmdir – Verzeichnis verschieben oder umbenennen

SYNTAX:

/etc/mmdir Verzeichnis Name

BESCHREIBUNG:

Mmdir verschiebt Verzeichnisse innerhalb eines Dateisystems (entspricht einer Umbenennung). Das angegebene Verzeichnis muß ein Verzeichnis sein; *Name* darf noch nicht angelegt sein. Keiner der angegebenen Namen darf eine Untermenge des anderen sein (/x/y kann nicht nach /x/y/z verschoben werden und auch nicht umgekehrt).

Nur der Superuser kann mmdir benutzen.

SIEHE AUCH:

Benutzerkommando: mkdir

Anhang 1: Administrator-Kommandos

ncheck – Namen aus I-Nummern generieren

SYNTAX:

```
/etc/ncheck [-i I-Nummern] [-a] [-s] [Dateisystem]
```

BESCHREIBUNG:

Ncheck ohne Argumente generiert eine Liste mit gegenübergestellten Pfadnamen und I-Nummern für alle Dateien in der Standardliste der Dateisysteme. Die Namen von Verzeichnisdateien werden mit Schrägstrich und Punkt (/.) abgeschlossen. Geben Sie die Option -i an, so werden nur die Dateien aufgelistet, deren I-Nummern auf die Option folgen. Mit der Option -a werden die Namen von . und .. ausgegeben, die normalerweise unterdrückt werden. Mit der Option -s wird die Auswertung auf Gerätedateien und Dateien mit Set-User-ID-Mode beschränkt. Die Absicht dabei ist, versteckte Brüche in den Sicherheitsmechanismen zu entdecken.

Es kann ein Dateisystem angegeben werden.

Die Reihenfolge der Auswertung ist nicht von vornherein sinnvoll. Daher sollte sie sortiert werden.

DIAGNOSE:

Wenn die Struktur des Dateisystems fehlerhaft ist, bezeichnet ?? den „Vater“ einer vaterlosen Datei, und ein mit ... beginnender Pfadname bedeutet eine Schleife.

SIEHE AUCH:

Administrator-Kommando: fsck

Benutzerkommando: sort

Anhang 1: Administrator-Kommandos

prfld, prfstat, prfdc, prfsnap, prfpr – Betriebssystem-Profiler

SYNTAX:

```
/etc/prfld [Namensliste]  
/etc/prfstat on  
/etc/prfstat off  
/etc/prfdc Datei [Periode [aus_Stunde]]  
/etc/prfsnap Datei  
/etc/prfpr Datei [Grenze [Namensliste]]
```

BESCHREIBUNG:

Prfld, prfstat, prfdc, prfsnap und prfpr bilden ein System von Programmen, mit deren Hilfe die Aktivitäten des UNIX-Betriebssystems ausgewertet werden können.

Prfld dient zum Initialisieren des Aufzeichnungsmechanismus im System. Es erzeugt eine Tabelle mit den Anfangsadressen aller System-Unterprogramme, die in der Namensliste enthalten sind.

Prfstat aktiviert und deaktiviert den Abfragemechanismus. Der Overhead der Profiler-Programme beträgt weniger als 1 % auf der Grundlage von 500 Textadressen. Prfstat gibt auch die Anzahl der ausgewerteten Textadressen aus.

Prfdc und **prfsnap** führen die Datenerfassungsfunktion für das Profilerprogramm aus, indem sie den aktuellen Wert aller Textadreßzähler in eine Datei kopieren, wo die Daten ausgewertet werden können. Prfdc legt die Zähler alle in *Periode* angegebenen Minuten in *Datei* ab und schaltet bei der angegebenen *aus_Stunde* aus (gültige Werte für *aus_Stunde* sind 0–24). Prfsnap erfaßt Daten nur beim Aufruf und fügt die Zählerwerte an *Datei* an.

Anhang 1: Administrator-Kommandos

Prfpr formatiert die von **prfdc** oder **prfsnap** erfaßten Daten. Alle Textadressen werden in das nächste Textsymbol (das in Namensliste gefunden wird) umgewandelt und ausgegeben, wenn die prozentuale Aktivität für diesen Bereich größer als *Grenze* ist.

DATEIEN:

`/dev/prf` Schnittstelle zu Profildaten und Textadressen
`nix` Standard-Namensliste

SIEHE AUCH:

Gerätedatei: `prf`

Anhang 1: Administrator-Kommandos

pwck, grpck – Prüfen der Paßwort- bzw. Gruppendatei

SYNTAX:

/etc/pwck [Datei]

/etc/grpck [Datei]

BESCHREIBUNG:

Pwck überprüft die Paßwortdatei und meldet gegebenenfalls gefundene Inkonsistenzen. Das Kommando überprüft die Anzahl der Felder, den Login-Namen, die Benutzer-ID, die Gruppen-ID und ob das Login-Verzeichnis und der wahlfreie Programmname existieren. Geben Sie keine andere Datei an, wird */etc/passwd* überprüft.

Grpck überprüft alle Einträge in der Gruppendatei. Es prüft die Anzahl der Felder, den Gruppennamen, die Gruppen-ID und ob alle Login-Namen auch in der Paßwortdatei eingetragen sind. Geben Sie keine Datei an, wird */etc/group* überprüft.

DATEIEN:

/etc/group

/etc/passwd

SIEHE AUCH:

Dateiformate: *group, passwd*

Anhang 1: Administrator-Kommandos

reject, accept – Annahme von Druckaufträgen sperren/freigeben

SYNTAX:

```
/usr/lib/reject [-r[Grund]] Ziele
/usr/lib/accept Ziele
```

BESCHREIBUNG:

Das **accept**-Kommando erlaubt dem Kommando lp für die angegebenen Ziele Druckaufträge anzunehmen. *Ziel* kann ein Drucker oder eine Klasse von Druckern sein. Mit Hilfe des Kommandos lpstat können Sie den Status der Ziele feststellen.

Durch **reject** untersagen Sie dem lp-Kommando Druckaufträge für die angegebenen Ziele anzunehmen. *Ziel* kann ein Drucker oder eine Klasse von Druckern sein. Mit lpstat können Sie den Status der Ziele feststellen.

Option:

-r[Grund] Angabe des Grundes für die Sperrung. Dieser Grund gilt für alle Drucker bis zur nächsten Option -r. Der Grund wird zum einen von lp ausgegeben, wenn ein Benutzer Aufträge für die gesperrten Ziele absetzt und zum anderen von lpstat. Geben Sie die Option -r nicht oder ohne *Grund* an, wird ein Standardtext verwendet.

DATEIEN:

```
/usr/spool/lp/*
```

SIEHE AUCH:

Administrator-Kommandos: lpadmin, lpsched
Benutzerkommandos: enable, lp, lpstat

Anhang 1: Administrator-Kommandos

runacct – Tägliche Abrechnungen ausführen

SYNTAX:

```
/usr/lib/acct/runacct [MMTT[Status]]
```

BESCHREIBUNG:

Runacct ist die hauptsächliche Shell-Prozedur für die tägliche Abrechnung. Sie wird normalerweise über cron gestartet. Runacct verarbeitet die Dateien für die Anschlußzeit-, Gebühren-, Platten- und Prozeßabrechnungen. Ferner erstellt die Prozedur Summendateien für prdaily bzw. die Fakturierung.

Runacct ist so ausgelegt, daß aktive Abrechnungsdateien oder Summendateien im Fehlerfall nicht in Mitleidenschaft gezogen werden. Es protokolliert sein Vorgehen, indem es beschreibende Diagnosemeldungen in die Datei active schreibt. Bei Erkennung eines Fehlers wird eine Meldung an /dev/console ausgegeben, eine Nachricht wird per mail an root und adm abgesetzt, und dann terminiert runacct. Runacct schützt sich durch eine Reihe von Sperrdateien vor dem erneuten Aufruf. Mit den Dateien lock und lock1 werden mehrere gleichzeitige Aufrufe verhindert, und mit lastdate wird das mehrmalige Aufrufen an ein und demselben Tag gesperrt.

Runacct teilt seine Verarbeitung in einzelne wiederanlauffähige Zustände ein, und der letzte abgeschlossene Status wird in statefile festgehalten. Zu diesem Zweck schreibt es den Status-Namen in statefile. Runacct schaut dann in statefile nach, was bereits erledigt ist und welcher Schritt als nächstes kommt. Die Stati werden in der folgenden Reihenfolge ausgeführt:

SETUP	Aktive Abrechnungsdateien in Arbeitsdateien verschieben (move).
WTMPFIX	Die Integrität der wtmp-Datei überprüfen und gegebenenfalls Datumsänderungen korrigieren.
CONNECT1	Sätze für Anschlußzeiten im ctmp.h-Format erstellen.
CONNECT2	Ctmp.h-Sätze in das tacct.h-Format konvertieren.

Anhang 1: Administrator-Kommandos

PROCESS	Prozeßabrechnungssätze in das tacct.h-Format konvertieren.
MERGE	Die Anschlußzeit- und Prozeßabrechnungssätze mischen.
FEES	Die Ausgabe von chargefee in das tacct.h-Format konvertieren und mit den Anschlußzeit- und Prozeßabrechnungssätzen mischen.
DISK	Plattenabrechnungssätze mit Anschlußzeit-, Prozeß- und Gebührenabrechnungssätzen mischen.
MERGETACCT	Die täglichen Summen-Abrechnungssätze in daytacct mit den zusammengefaßten Summen-Abrechnungssätzen in /usr/adm/acct/sum/tacct mischen.
CMS	Zusammenfassende Werte für Kommandos erstellen.
USEREXIT	Hier können gegebenenfalls anlagenabhängige Abrechnungsprogramme angeschlossen werden.
CLEANUP	Temporäre Dateien löschen und terminieren.

Soll runacct nach einem Fehler neu gestartet werden, prüfen Sie zunächst, ob Diagnosemeldungen in der Datei active vorhanden sind. Sodann korrigieren Sie verfälschte Datendateien wie etwa pacct oder wtmp. Die Sperrdateien (lock) und die Datei lastdate müssen gelöscht werden, bevor runacct neu gestartet werden kann. Zum Neustart von runacct muß das Argument *MMTT* unbedingt angegeben werden. Es spezifiziert den Monat und den Tag, für den runacct die Abrechnung wiederholen soll. Die Einsprungstelle für die Verarbeitung hängt vom Inhalt von statefile ab; soll dies anders sein, so muß der gewünschte Status, der den Anfangspunkt der Verarbeitung angibt, in der Kommandozeile angegeben werden.

Anhang 1: Administrator-Kommandos

BEISPIELE:

1. Starten von runacct:
nohup runacct 2> /usr/adm/acct/nite/fd2log &
2. Neustart von runacct:
nohup runacct 0601 2>> /usr/adm/acct/nite/fd2log &
3. Neustart von runacct bei einem bestimmten Status:
nohup runacct 0601 MERGE 2>> /usr/adm/acct/nite/fd2log &

DATEIEN:

/etc/wtmp
/usr/adm/pacct*
/usr/src/cmd/acct/tacct.h
/usr/src/cmd/acct/ctmp.h
/usr/adm/acct/nite/active
/usr/adm/acct/nite/dayacct
/usr/adm/acct/nite/lock
/usr/adm/acct/nite/lock1
/usr/adm/acct/nite/lastdate
/usr/adm/acct/nite/statefile
/usr/adm/acct/nite/ptacct*.mmtt

HINWEIS:

Oft ist es gefährlich, runacct im SETUP-Status neu zu starten. SETUP sollte manuell ausgeführt und runacct wie folgt neu gestartet werden:

```
runacct MMTT WTMPFIX
```

Hat runacct im PROCESS-Status terminiert, löschen Sie die letzte ptacct-Datei, weil diese unvollständig ist.

Anhang 1: Administrator-Kommandos

SIEHE AUCH:

Administrator-Kommandos: acct, acctcms, acctcon, acctmerg, acctprc,
acctsh, cron, fwtmp

Dateiformate: acct, utmp

Benutzerkommandos: acctcom, mail

Systemschnittstelle: acct

Anhang 1: Administrator-Kommandos

sadp – Auswertungsprogramm für Plattenzugriffe

SYNTAX:

```
sadp [-th] [-d Gerät[-Laufwerk]] s [n]
```

BESCHREIBUNG:

Sadp gibt eine Auswertung der Adressen und Positionierdistanzen bei Plattenzugriffen in tabellarischer Form oder als Histogramm aus. Es fragt die Plattenaktivität einmal pro Sekunde während eines Zeitraums von *s* Sekunden ab. Dies geschieht wiederholt, wenn *n* angegeben ist. Die Zylinderbelegung und die Positionierdistanzen werden in Einheiten von 8 Zylindern aufgezeichnet.

Gültige Werte für *Gerät* sind rp06, rm05 und disk. *Laufwerk* gibt die Plattenlaufwerke an und kann folgende Werte haben:

1. eine Laufwerksnummer in dem für *Gerät* zulässigen Bereich, zwei durch ein Minuszeichen verbundene Nummern (die einen Bereich von – bis einschließlich angeben) oder
2. eine Liste von durch Kommata getrennten Laufwerksnummern.

Bis zu 8 Plattenlaufwerke können ausgewertet werden. Die Option -d kann fortfallen, wenn nur ein Gerät vorhanden ist.

Geben Sie die Option -t an, so werden die Daten in tabellarischer Form ausgegeben. Geben Sie -h an, so werden die Daten als Histogramm ausgedruckt. Standard ist -t.

BEISPIEL:

```
sadp -d rp06-0 900 4
```

erzeugt 4 tabellarische Auswertungen, die jeweils die Zylinderbelegung und die Positionierdistanz während eines Zeitraums von 15 Minuten beim Gerät rp06, Plattenlaufwerk 0 beschreiben.

DATEIEN:

```
/dev/kmem
```

Anhang 1: Administrator-Kommandos

sa1, sa2, sadc – Auswertung der Systemaktivitäten

SYNTAX:

```
/usr/lib/sa/sadc [t n] [Ausgabedatei]
/usr/lib/sa/sa1 [t n]
/usr/lib/sa/sa2 [-ubdycwaqvmA] [-s Zeit] [-e Zeit] [-i Sek]
```

BESCHREIBUNG:

Daten über die Systemaktivitäten können auf besondere Anforderung eines Benutzers (siehe Benutzerkommando sar) – aber auch automatisch und routinemäßig wie hier beschrieben – abgerufen werden. Das Betriebssystem führt eine Reihe von Zählern, die bei bestimmten Systemaktivitäten hochgezählt werden. Es handelt sich um Zähler für die CPU-Auslastung, die Pufferbelegung, die E/A-Aktivitäten bei Platten und Bändern, die Aktivität von Terminals, Umschaltungen und Systemaufrufe, Dateizugriffe, die Warteschlangen-Aktivität und die Interprozess-Kommunikation.

Mit sadc und den Shell-Prozeduren sa1 und sa2 werden diese Daten abgefragt, gespeichert und verarbeitet.

Das Datensammelkommando **sadc** fragt die Systemdaten *n* mal alle *t* Sekunden ab und gibt sie im Binärformat in die angegebene Ausgabedatei oder auf der Standardausgabe aus. Geben Sie *t* und *n* nicht an, wird ein spezieller Satz ausgegeben. Diese Funktion dient dazu, beim Booten des Systems den Zeitpunkt zu markieren, an dem die Zähler wieder bei 0 beginnen. Der Eintrag

```
su sys -c "/usr/lib/sa/sadc /usr/adm/sa/sa'date +%d"
```

in /etc/rc schreibt den speziellen Satz in die tägliche Datendatei, um den Neustart des Systems zu kennzeichnen.

Anhang 1: Administrator-Kommandos

Das Shell-Script **sa1** ist eine Variante von `sadc` und dient dazu, die Daten zu erfassen und in der Binärdatei `/usr/adm/sa/satt` zu speichern, wobei `tt` der aktuelle Tag ist. Die Argumente `t` und `n` bewirken, daß Sätze `n` mal in Intervallen von `t` Sekunden geschrieben werden; sind `t` und `n` weggelassen, erfolgt die Ausgabe nur einmal.

Die Einträge:

```
0 * * * 0,6 /usr/lib/sa/sa1
0 8-17 * * 1-5 /usr/lib/sa/sa1 1200 3
0 18-7 * * 1-5 /usr/lib/sa/sa1
```

in `/usr/spool/cron/crontab/sys` geben während der Arbeitszeit alle 20 Minuten und sonst jede Stunde einmal die Datensätze aus.

Das Shell-Script **sa2**, eine Variante von `sar`, schreibt eine tägliche Auswertung in die Datei `/usr/adm/sa/sartt`. Die Optionen sind unter dem Benutzerkommando `sar` erläutert. Der crontab-Eintrag

```
5 18 * * 1-5 /usr/lib/sa/sa2 -s 8:00 -e 18:01 -i 3600 -A
```

meldet wichtige Aktivitäten während der Arbeitszeit stündlich.

Anhang 1: Administrator-Kommandos

Die binäre tägliche Datendatei hat folgende Struktur:

```

struct sa
{
    struct sysinfo si;      /* see /usr/include/sys/sysinfo.h */
    int  szi-node;         /* current entries of i-node table */
    int  szfile;           /* current entries of file table */
    int  sztext;           /* current entries of text table */
    int  szproc;           /* current entries of proc table */
    int  mszi-node;        /* size of i-node table */
    int  mszfile;          /* size of file table */
    int  msztext;          /* size of text table */
    int  mszproc;          /* size of proc table */
    long i-nodeovf;        /* cumul. overflows of i-node table */
    long fileovf;          /* cumul. overflows of file table */
    long textovf;          /* cumul. overflows of text table */
    long procovf;          /* cumul. overflows of proc table */
    time_t ts;             /* time stamp, seconds */
    long devio[NDEVS] [4]; /* device info for up to NDEVS units */
#define IO_OPS      0 /* cumul. I/O requests */
#define IO_BCNT     1 /* cumul. blocks transferred */
#define IO_ACT      2 /* cumul. drive busy time in ticks */
#define IO_RESP     3 /* cumul. I/O resp time in ticks */
};
    
```

DATEIEN:

```

/usr/adm/sa/satt    Tägliche Datendatei
/usr/adm/sa/sartt   Tägliche Auswertungsdatei
/tmp/sa.adrfl      Adreßdatei
    
```

SIEHE AUCH:

Administrator-Kommando: cron
Benutzerkommandos: sar, timex

A1

..Weitergabe sowie Vervielfältigung dieser Unterlagen, Verwertung und
 Mitteilung ihres Inhalts nicht gestattet, soweit nicht ausdrücklich zugestanden.
 Zuwendungsbedingungen verpflichten zu Schadensersatz. Alle Rechte für den Fall
 der Patenterteilung oder Gebrauchsmustererfindung vorbehalten.

Anhang 1: Administrator-Kommandos

setmnt – Mount-Tabelle einrichten

SYNTAX:

```
/etc/setmnt
```

BESCHREIBUNG:

Setmnt legt die Tabelle /etc/mnttab an, die für die Administrator-Kommandos mount und umount benötigt wird. Setmnt liest die Standard-eingabe und legt pro Zeile einen mnttab-Eintrag an. Die Eingabezeilen haben die folgende Form:

```
Dateisystem Knoten
```

wobei *Dateisystem* der Name der Gerätedatei des Dateisystems (z. B. nd02) und *Knoten* der Name der Wurzel dieses Dateisystems (z. B. /usr) ist.

DATEIEN:

```
/etc/mnttab
```

HINWEIS:

Größere Probleme treten auf, wenn *Dateisystem* oder *Knoten* länger als 32 Zeichen sind. Setmnt begrenzt die Anzahl der mnttab-Einträge ohne Meldung.

SIEHE AUCH:

Administrator-Kommando: mount

Dateiformat: mnttab

Anhang 1: Administrator-Kommandos

shutdown – Beenden der Verarbeitung

SYNTAX:

/etc/shutdown

BESCHREIBUNG:

Shutdown versetzt das System in den Single-User-Modus. Insbesondere beendet shutdown die Dämonprozesse des Accounting-Systems und alle laufenden Prozesse in geordneter Form.

Die shutdown-Prozedur führt einen Dialog mit dem Bediener (d. h. der Person, die shutdown aufgerufen hat). Shutdown weist den Bediener unter Umständen an, bestimmte Maßnahmen zu treffen oder bestimmte Antworten einzugeben, bevor die Ausführung fortgesetzt werden kann.

Shutdown geht in folgenden Schritten vor:

1. Alle im System angemeldeten Benutzer werden durch eine Mitteilung aufgefordert, sich abzumelden. Der Administrator kann eine eigene Meldung senden oder es wird die übliche Meldung zur Dateisicherung angezeigt.
2. Will der Administrator eine Datensicherung durchführen, hängt shutdown alle Dateisysteme aus.
3. Die Superblöcke aller Dateisysteme werden aktualisiert, bevor das System gestoppt werden soll. Dies muß vor dem erneuten Booten durchgeführt werden, um die Integrität der Dateisysteme zu gewährleisten.

MELDUNGEN:

Die häufigste Fehlermeldung ist „device busy“. Diese Meldung erfolgt, wenn ein bestimmtes Dateisystem nicht ausgehängt werden konnte.

SIEHE AUCH:

Administrator-Kommando: mount
Benutzerkommando: sync

Anhang 1: Administrator-Kommandos

sysdef – Systemdefinition

SYNTAX:

```
/etc/sysdef [Betriebssystem [Master]]
```

BESCHREIBUNG:

Sysdef analysiert die angegebene Betriebssystemdatei und holt sich aus ihr Konfigurationsinformationen. Hierzu zählen alle Hardware-Geräte sowie Systemgeräte und alle veränderbaren Parameter.

DATEIEN:

/unix Standard-Namensliste

/etc/master Standardmäßige Tabelle für Hardwarespezifikationen

HINWEIS:

Da das System keine Informationen über die Aliasnamen von config speichert, sind die zurückgelieferten Gerätenamen unter Umständen falsch.

Anhang 1: Administrator-Kommandos

tic – Termino-Compiler

SYNTAX:

tic [-v[n]] *Datei* ...

BESCHREIBUNG:

Tic kompiliert terminfo-Quelldateien. Das Ergebnis wird im Verzeichnis /usr/lib/terminfo abgelegt.

Geben Sie die Option -v an, so gibt tic Trace-Informationen über seinen Verlauf aus. Mit der optionalen Zahl *n* kann der Grad der Ausführlichkeit gesteigert werden.

Tic kompiliert alle terminfo-Beschreibungen in den angegebenen Dateien. Wenn ein Feld use= gefunden wird, durchsucht tic zunächst die aktuelle Datei und dann die Masterdatei, also „./terminfo.src“.

Ist die Umgebungsvariable TERMINFO gesetzt, werden die Ergebnisse dort abgelegt statt in /usr/lib/terminfo.

HINWEIS:

Die kompilierten Einträge dürfen insgesamt nicht größer als 4096 Bytes sein. Der Eintrag im Namensfeld darf 128 Bytes nicht übersteigen.

Tic durchsucht ./terminfo.src, statt zu prüfen, ob ein kompilierter Eintrag vorliegt.

DATEIEN:

/usr/lib/terminfo/*/* kompilierte Datenbasis der Terminal-Funktionsmöglichkeiten

Anhang 1: Administrator-Kommandos

umount, mount – Dateisystem aus- und einhängen

SYNTAX:

```
/etc/umount Name  
/etc/mount [Name Verzeichnis[-r]]
```

BESCHREIBUNG:

Name bezeichnet die Gerätedatei einer logischen Platte (Major-Nummer 107). *Verzeichnis* wird der Name des Root-Verzeichnisses des montierten Dateisystems. Ein Verzeichnis mit diesem Namen muß im Root-Dateisystem angelegt sein.

Soll das montierte Dateisystem nur lesbar sein, so ist die Option *-r* anzugeben. Die Dateisysteme können nur auf Platten betrieben werden, die nicht physikalisch schreibgeschützt sind.

Mount und **umount** verwalten eine Mount-Tabelle. Rufen Sie **mount** ohne Parameter auf, dann wird diese Tabelle ausgedruckt.

Umount demontiert das Dateisystem der logischen Platte *Name*.

HINWEIS:

Bevor Sie **mount** aufrufen, sollten Sie das entsprechende Dateisystem mit **fsck** überprüfen.

MELDUNGEN:

Mount gibt eine Warnung aus, wenn das angegebene Verzeichnis nicht mit dem Verzeichnis übereinstimmt, auf das das Dateisystem zuvor montiert war.

Umount wird nicht ausgeführt, wenn das Dateisystem aktiv ist. Ein Dateisystem ist aktiv, wenn eine Datei eröffnet ist oder ein Benutzer im Dateisystem arbeitet.

Anhang 1: Administrator-Kommandos

DATEIEN:

/etc/mnttab Mount-Tabelle

SIEHE AUCH:

Administrator-Kommandos: mkfs, fsck

Dateiformat: mnttab

Systemschnittstelle: mount

Anhang 1: Administrator-Kommandos

uuclean – Bereinigung des Spool-Verzeichnisses von uucp

SYNTAX:

`/usr/lib/uucp/uuclean [Optionen]`

BESCHREIBUNG:

Uuclean durchsucht das Spool-Verzeichnis nach Dateien mit dem angegebenen Präfix und löscht alle Dateien, die älter sind als die angegebene Anzahl Stunden.

Optionen:

- `-dVerzeichnis` Es soll das angegebene Verzeichnis anstelle des Spool-Verzeichnisses bereinigt werden. Ist das Verzeichnis kein gültiges Spool-Verzeichnis, kann es keine „Arbeitsdateien“ enthalten, d. h. Dateien, deren Namen mit C. beginnen. Diese Dateien haben für uuclean eine besondere Bedeutung und beziehen sich auf Job-Statistiken von uucp.
- `-p[Präfix]` Sucht nach Dateien, die mit dem angegebenen Präfix beginnen. Bis zu zehn Argumente `-p` können angegeben werden. Ein `-p` ohne anschließendes Präfix bewirkt, daß alle Dateien, die älter sind als die angegebene Zeit, gelöscht werden.
- `-nStunden` Dateien, die älter sind als die angegebene Anzahl Stunden, werden gelöscht, sofern das Präfix übereinstimmt. (Standardzeit ist 72 Stunden.)
- `-w[Datei]` Standardmäßig löscht uuclean Dateien, die älter sind als eine angegebene Anzahl Stunden (siehe Option `-n`). Mit der Option `-w` werden die Dateien, die älter sind als die angegebene Anzahl Stunden, lediglich ermittelt, aber nicht gelöscht. Ist das Argument *Datei* angegeben, so werden die ermittelten Namen in diese Datei geschrieben, andernfalls auf der Standardausgabe angezeigt.

Anhang 1: Administrator-Kommandos

- sSystem** Es werden nur Dateien untersucht, die für das System bestimmt sind. Die Option -s kann bis zu zehn mal wiederholt werden.
- m[Datei]** Wird die Option -m angegeben, so wird der Eigentümer der zu löschenden Dateien per mail über den Löschvorgang benachrichtigt. Ist *Datei* angegeben, wird ein Eintrag in dieser Datei abgelegt.

Uclean wird im typischen Fall über cron gestartet.

DATEIEN:

- /usr/lib/uucp** Verzeichnis mit Kommandos, die von uclean intern benutzt werden.
- /usr/spool/uucp** Spool-Verzeichnis

SIEHE AUCH:

Administrator-Kommando: cron
Benutzerkommandos: uucp, uux

Anhang 1: Administrator-Kommandos

uusub – Überwachung eines uucp-Netzes

SYNTAX:

```
/usr/lib/uucp/uusub [Optionen]
```

BESCHREIBUNG:

Uusub definiert ein uucp-Teilnetz und überwacht die Netzverbindungen und den Datenverkehr unter den Mitgliedern des Teilnetzes. Es sind folgende Optionen zulässig:

- a*System* *System* wird in das Teilnetz (Liste der zu überwachenden Systeme) aufgenommen.
- d*System* *System* wird aus dem Teilnetz (Liste der zu überwachenden Systeme) entfernt.
- l Statistische Daten über Verbindungen ausgeben.
- r Statistische Daten über Verkehrsaufkommen ausgeben.
- f Statistiken über Verbindungen löschen.
- u*Stunden* Die Verkehrsstatistiken für die angegebene Anzahl Stunden erfassen.
- c*System* Die Verbindung zu dem angegebenen System wird überprüft. Wird *System* mit all spezifiziert, wird die Verbindung zu allen zu überwachenden Systemen im Teilnetz überprüft.

Die Angaben in der Auswertung über die Verbindungen haben folgende Bedeutung:

```
sys #call #ok time #dev #login #nack #other
```

Dabei ist sys der Name des Nachbarsystems, #call gibt an, wie oft das lokale System seit der letzten Löschung der Statistiken versucht hat, sys anzurufen, #ok ist die Anzahl der erfolgreichen Verbindungen, time ist die Zeit der letzten erfolgreichen Verbindung, #dev ist die Anzahl der erfolglosen Verbindungsversuche, weil kein Gerät verfügbar war (z. B. automatische Anrufeinrichtung), #login ist die Anzahl der erfolglosen Verbindungsversuche wegen negativ verlaufener Anmeldung,

Anhang 1: Administrator-Kommandos

#nack ist die Anzahl der erfolglosen Verbindungsversuche wegen fehlender Antwort (z. B. Leitung belegt, System ausgefallen), und #other ist die Anzahl der erfolglosen Verbindungsversuche wegen anderer Ursachen.

Die Felder der Verkehrsstatistik haben folgende Bedeutung:

sfile sbyte rfile rbyte

Dabei bedeutet sfile die Anzahl der Dateien und sbyte die Anzahl der Bytes, die in dem Zeitraum gesendet wurden, der von dem letzten uucsub-Kommando mit der Option -uStunden angegeben wird. Analog sind rfile und rbyte die Anzahl der empfangenen Dateien bzw. Bytes.

Das Kommando:

uusub -c all -u 24

wird von cron normalerweise einmal täglich gestartet.

DATEIEN:

/usr/spool/uucp/SYSLOG	System-Protokolldatei
/usr/lib/uucp/Lsub	Verbindungsstatistik
/usr/lib/uucp/Rsub	Verkehrsstatistik

SIEHE AUCH:

Benutzerkommandos: uucp, uustat

Anhang 1: Administrator-Kommandos

volcopy, labelit – Dateisysteme unter Kennsatzprüfung kopieren

SYNTAX:

```
/etc/volcopy [Optionen] Dateisystem Gerät1 Volname1 Gerät2 Volname2
```

```
/etc/labelit Gerät [Dateisystem Datenträger [-n]]
```

BESCHREIBUNG:

Volcopy fertigt eine genaue Kopie des angegebenen Dateisystems mit einer Blocklänge an, die auf das angegebene Gerät abgestimmt ist.

Folgende Optionen stehen zur Verfügung:

- a Anstelle der üblichen 10 Sekunden langen Wartezeit wird eine Abfrage-Routine aufgerufen, die vom Bediener positiv beantwortet werden muß, bevor die Kopie angefertigt wird.
- s (Standard) Bei falscher Identifikation wird DEL aufgerufen.

Weitere Optionen beziehen sich nur auf Magnetbänder:

- bpi*Dichte* Aufzeichnungsdichte in Bits pro Inch (d. h. 800/1600/6250).
- feet*Länge* Länge des Bandes in Fuß (d. h. 1200/2400).
- reel*Num* Erste Bandnummer für eine neu gestartete Kopie.
- buf Mit doppelt gepufferter E/A arbeiten.

Das Programm fragt nach der Länge und Aufzeichnungsdichte, wenn diese Angaben nicht in der Kommandozeile stehen und nicht in dem Kennsatz eines Eingabebandes verzeichnet sind. Ist das Dateisystem so groß, daß es nicht auf ein Band paßt, verlangt volcopy weitere Bänder. Die Kennsätze aller Bänder werden geprüft. Die Bänder können abwechselnd in zwei oder mehr Laufwerke eingelegt werden.

Dateisystem ist der Name (z. B. root, u1 usw.) des zu kopierenden Dateisystems.

Anhang 1: Administrator-Kommandos

Gerät ist der physikalische Plattenabschnitt bzw. das Band (z. B. /dev/rmt0 usw).

Volname ist der physikalische Datenträgername und sollte mit dem Namen auf dem Klebeetikett des Datenträgers übereinstimmen. Der Name darf maximal sechs Stellen lang sein. *Volname* kann als – angegeben werden, wenn der vorhandene Datenträgername verwendet werden soll.

Gerät1 und *Volname1* sind das Gerät und der Datenträger, von dem die Kopie des Dateisystems übernommen wird. *Gerät2* und *Volname2* sind das Zielgerät bzw. der Zieldatenträger.

Dateisystem und *Volname* werden in den letzten 12 Stellen des Superblocks protokolliert (char ffname[6], volname[6];).

Mit **labelit** können ausgehängten Dateisystemen auf Platte oder Band Kennsätze zugewiesen werden. Lassen Sie die optionalen Argumente weg, gibt labelit die aktuellen Kennsatzwerte aus.

Die Option -n initialisiert Kennsätze auf Bändern (dadurch wird der bisherige Inhalt zerstört).

DATEIEN:

/etc/log/filesave.log Protokoll der kopierten Dateisysteme/Datenträger

HINWEIS:

Nur solche Gerätenamen, die mit /dev/rmt beginnen, werden als Magnetbänder behandelt.

SIEHE AUCH:

Dateiformat: fs

Anhang 1: Administrator-Kommandos

wall – Rundschreiben an alle Benutzer senden

SYNTAX:

```
/etc/wall
```

BESCHREIBUNG:

Wall liest die Standardeingabe bis zu einem EOF-Zeichen. Daraufhin wird diese Nachricht an alle zur Zeit im System angemeldeten Benutzer gesendet.

Hauptsächlich dient wall dazu, die Benutzer über systemspezifische Angelegenheiten zu unterrichten.

Wall sollte nur vom Superuser aufgerufen werden, um einen eventuell gesetzten Schreibschutz der Terminals zu brechen.

DATEIEN:

```
/dev/tty*
```

SIEHE AUCH:

Benutzerkommandos: mesg, write

Anhang 1: Administrator-Kommandos

whodo – Wer tut was

SYNTAX:

/etc/whodo

BESCHREIBUNG:

Whodo erstellt eine gemischte, umformatierte und mit Datum versehene Ausgabe der Kommandos who und ps.

DATEIEN:

etc/passwd

SIEHE AUCH:

Benutzerkommandos: ps, who

Anhang 2: Gerätedateien

A2 Gerätedateien

err – Fehlerprotokollierungs-Schnittstelle

BESCHREIBUNG:

Das Minor-Gerät 0 des err-Treibers ist die Schnittstelle zwischen einem Prozeß und den Fehlersatz-Sammelroutinen des Systems. Der Treiber kann nur für das Lesen durch einen einzelnen Prozeß mit Superuser-Zugriffsrechten eröffnet werden. Bei jeder Leseoperation wird ein vollständiger Fehlersatz ausgelesen; der Satz wird abgeschnitten, wenn in dem Leseauftrag weniger als die Satzlänge angegeben ist.

DATEIEN:

/dev/error Gerätedatei

SIEHE AUCH:

Administrator-Kommando: errdemon

Anhang 2: Gerätedateien

mem, kmem – Hauptspeicher-Abbild

BESCHREIBUNG:

Mem ist eine Gerätedatei, die ein Abbild des Hauptspeichers des Computers darstellt. Mit ihrer Hilfe kann beispielsweise das System untersucht und sogar mit Patches korrigiert werden.

Die Byte-Adressen in **mem** werden als Hauptspeicheradressen interpretiert. Werden nicht-existente Adressen angesprochen, so wird ein Fehler zurückgemeldet.

Die Datei **kmem** entspricht der Datei **mem**, wobei jedoch der virtuelle Speicher im Kernel statt dem physikalischen Speicher angesprochen wird.

DATEIEN:

`/dev/mem`

`/dev/kmem`

Anhang 2: Gerätedateien

null – Die Null-Datei

BESCHREIBUNG:

In eine Null-Gerätedatei geschriebene Daten werden weggeworfen.

Bei einer Leseanforderung aus einer Null-Gerätedatei werden immer 0 Bytes gelesen.

DATEIEN:

/dev/null

Anhang 2: Gerätedateien

prf – Betriebssystem-Profiler

BESCHREIBUNG:

Die Datei prf enthält Informationen über Aktivitäten im Betriebssystem. Beim Beschreiben der Datei wird die Auswertungsfunktion mit den zu überwachenden Textadressen geladen. Wenn die Datei gelesen wird, erhält man diese Adressen und eine Reihe von Zählern, die Aufschluß über die Aktivität zwischen benachbarten Textadressen geben.

Der Aufzeichnungsmechanismus wird von der Systemuhr gesteuert und fragt den Programmzähler ab. Die Abfragen, bei denen festgestellt wird, daß das Betriebssystem aktiv ist, werden mit den gespeicherten Textadressen verglichen und erhöhen entsprechende Zähler für die spätere Verarbeitung.

Die Datei prf ist ein Pseudogerät ohne zugehörige Hardware.

DATEIEN:

/dev/prf

SIEHE AUCH:

Administrator-Kommando: profiler

Anhang 2: Gerädateien

sxt – Treiber für Pseudogeräte

BESCHREIBUNG:

Sxt ist ein Treiber für Pseudogeräte, der zwischen die üblichen Terminal-Leitungsprotokolle und einem realen Gerätetreiber ein weiteres Protokoll einschiebt. Die Standardprotokolle bearbeiten virtuelle Terminal-Strukturen (Kanäle), die vom sxt-Treiber deklariert werden. Sxt wirkt als Protokoll, das eine reale Terminalstruktur bearbeitet, die von einem realen Gerätetreiber deklariert ist. Der sxt-Treiber wird derzeit nur vom Benutzerkommando shl benutzt.

Virtuelle Terminals werden von I-Knoten im Unterverzeichnis /dev/sxt benannt und werden in Gruppen von bis zu acht zugeordnet. Um eine Gruppe zuzuordnen, muß ein Programm eine Datei mit einem Namen der Form /dev/sxt/???0 (Kanal 0) exklusiv eröffnen und dann einen ioctl-Aufruf SXTIOCLINK ausführen, um das Multiplexen einzuleiten.

Es kann jeweils immer nur ein Kanal, und zwar der steuernde Kanal, Eingabedaten von der Tastatur empfangen; Leseversuche von anderen Kanälen werden blockiert.

Sxt unterstützt zwei Gruppen von ioctl-Kommandos. Die erste Gruppe enthält die normalen ioctl-Kommandos, die in termio beschrieben sind, und zusätzlich die beiden folgenden:

TIOEXCL Exklusiven Modus setzen: es sind keine weiteren opens zugelassen, bis die Datei geschlossen wurde.

TIOCNXCL Exklusiven Modus zurücksetzen: weitere opens sind wieder zulässig.

Die zweite Gruppe besteht in Direktiven an sxt selbst. Einige davon können nur auf Kanal 0 ausgeführt werden.

SXTIOCLINK Eine Kanalgruppe zuordnen und die virtuellen Terminals im Multiplexverfahren auf dem realen Terminal zusammenführen. Das Argument ist die Anzahl der zuzuordnenden Kanäle. Dieses Kommando kann nur auf Kanal 0 ausgeführt werden. Mögliche Fehler:

Anhang 2: Gerätedateien

EINVAL	Das Argument ist außerhalb des zulässigen Bereichs.
ENOTTY	Das Kommando wurde nicht von einem realen Terminal abgesetzt.
ENXIP	Linesw ist nicht bei sxt konfiguriert.
EBUSY	Es wurde bereits ein Kommando SXTIOCLINK für dieses reale Terminal abgesetzt.
ENOMEN	Es ist kein Systemspeicher für die Zuordnung der virtuellen Terminal-Strukturen verfügbar.
EBADF	Vor diesem Aufruf wurde Kanal 0 nicht eröffnet.
SXTIOCSWTCH	Den steuernden Kanal festlegen. Mögliche Fehler:
EINVAL	Es wurde eine ungültige Kanalnummer angegeben.
EPERM	Das Kommando wurde nicht von Kanal 0 ausgeführt.
SXTIOCWF	Veranlassen, daß ein Kanal wartet, bis er der steuernde Kanal ist. Dieses Kommando liefert den Fehler EINVAL zurück, wenn eine ungültige Kanalnummer angegeben wird.
SXTIOCUBLK	Das Steuer-Flag loblk im virtuellen Terminal des angegebenen Kanals ausschalten. Der Fehler EINVAL wird zurückgeliefert, wenn eine ungültige Kanalnummer oder Kanal 0 angegeben wird.
SXTIOCSTAT	Den Status (bei Eingabe oder Ausgabe geblockt) jedes Kanals feststellen und in der vom Argument angegebenen sxtblock-Struktur ablegen. Der Fehler EFAULT wird zurückgeliefert, wenn nicht in die Struktur geschrieben werden kann.
SXTIOCTRACE	Trace aktivieren. Dieses Kommando ist wirkungslos, wenn kein Trace konfiguriert ist.

Anhang 2: Gerätedateien

SXTIOCNOTRACE

Trace deaktivieren. Dieses Kommando ist wirkungslos, wenn keine Trace konfiguriert ist.

DATEIEN:

/dev/sxt/??[0-7] Virtuelle Terminals
/usr/include/sys/sxt.h Treiber-spezifische Definitionen

SIEHE AUCH:

Benutzerkommandos: shl, stty
Gerätedatei: termio
Systemschnittstellen: ioctl, open

Anhang 2: Gerätedateien

termio – Allgemeine Terminalschnittstelle

BESCHREIBUNG:

Alle asynchronen Kommunikationsanschlüsse arbeiten mit der selben allgemeinen Schnittstelle, gleichgültig um welche Hardware es sich handelt. Im weiteren Verlauf dieses Abschnitts werden die gemeinsamen Merkmale dieser Schnittstelle behandelt.

Wird eine Terminaldatei eröffnet, läßt sie den Prozeß normalerweise warten, bis eine Verbindung hergestellt ist. In der Praxis werden diese Dateien selten von Benutzerprogrammen eröffnet; sie werden von `getty` eröffnet und werden zur Standardeingabe, Standardausgabe und Standardfehlerausgabe des Benutzers. Die erste Terminaldatei, die vom führenden Prozeß einer Prozeßgruppe eröffnet wird und noch nicht einer anderen Prozeßgruppe zugeordnet ist, wird das Steuerterminal für diese Prozeßgruppe. Das Steuerterminal spielt bei der Behandlung von Quit- und Interrupt-Signalen eine besondere Rolle. Diese wird weiter unten behandelt. Das Steuerterminal wird bei einem `fork`-Systemaufruf von einem Kindprozeß geerbt. Ein Prozeß kann diese Zuordnung abändern, indem er seine Prozeßgruppe mit Hilfe des Systemaufrufs `setpgrp` ändert.

Ein Terminal, das einer dieser Dateien zugeordnet ist, arbeitet in der Regel im Vollduplexbetrieb. Es können jederzeit Zeichen eingegeben werden, auch während gleichzeitig Daten ausgegeben werden. Zeichen gehen nur verloren, wenn die Zeicheneingabepuffer des Systems voll sind (selten), oder wenn die maximal zulässige Anzahl Eingabezeichen erreicht ist, die noch nicht von einem Programm gelesen wurde. Derzeit liegt diese Grenze bei 256 Zeichen. Ist sie erreicht, werden alle gespeicherten Zeichen ohne Meldung weggeworfen.

Anhang 2: Gerätedateien

Normalerweise werden Terminaleingaben zeilenweise verarbeitet. Eine Zeile wird durch ein Neue Zeile-Zeichen (ASCII LF), ein Dateiende-Zeichen (ASCII EOT) oder ein Zeilenende-Zeichen beendet. Dies bedeutet, daß ein lesendes Programm warten muß, bis eine vollständige Zeile eingegeben ist. Ferner wird maximal eine Zeile übergeben, gleichgültig wie viele Zeichen im Leseaufruf angefordert wurden. Es ist jedoch nicht notwendig, eine ganze Zeile auf einmal zu lesen. In einem Lesevorgang können beliebig viele Zeichen, z. B. auch ein einzelnes Zeichen, angefordert werden, ohne daß Daten verlorengehen.

Während der Eingabe werden Löschvorgänge (erase für Zeichen und kill für Zeilen) normal verarbeitet. Standardmäßig löscht das Zeichen # das letzte angegebene Zeichen, es löscht aber nicht über den Anfang einer Zeile hinaus. Das Zeichen @ löscht die gesamte Eingabezeile und gibt wahlweise ein Neue Zeile-Zeichen aus. Beide Zeichen arbeiten auf Key-Stroke-Basis, d. h. unabhängig von möglicherweise durchgeführten Rückschritten und Tabsprüngen. Die Zeichen für Zeichenlöschung (erase) und Zeilenlöschung (kill) können in der angegebenen Form eingegeben werden, wenn ihnen ein Escape-Zeichen (\) vorangestellt wird. In diesem Fall wird das Escape-Zeichen nicht gelesen. Die Zeichen für Zeichenlöschung und Zeilenlöschung können geändert werden.

Bestimmte Zeichen haben bei der Eingabe eine Sonderfunktion. Nachfolgend finden Sie diese Funktionen mit ihren Standardwerten:

- | | |
|------|--|
| INTR | (Rubout oder ASCII DEL) erzeugt ein Interrupt-Signal, das an alle Prozesse des betreffenden Steuerterminals gesendet wird. Normalerweise werden alle diese Prozesse zwangsweise beendet, aber man kann durch entsprechende Vorkehrungen dafür sorgen, daß das Signal ignoriert wird oder daß das Signal abgefangen und eine positionsabhängige Maßnahme getroffen wird; siehe den Systemaufruf signal. |
| QUIT | (Control-I oder ASCII FS) erzeugt ein Quit-Signal. Es wird genauso behandelt wie das Interrupt-Signal. Hat allerdings ein empfangender Prozeß andere Vorkehrungen getroffen, wird er nicht nur beendet, sondern es wird auch eine Core-Image-Datei (mit dem Namen core) im aktuellen Verzeichnis angelegt. |

Anhang 2: Gerätedateien

SWTCH	(Control-z or ASCII SUB) wird von shl benutzt, um den aktuellen Layer in den Kontroll-Layer umzuwandeln.
ERASE	(#) löscht das vorangehende Zeichen. Es löscht nicht über den Anfang einer Zeile hinaus, die durch eines der Zeichen NL, EOF oder EOL begrenzt ist.
KILL	(@) löscht die gesamte Zeile, die von einem der Zeichen NL, EOF oder EOL beendet wird.
EOF	(Control-d oder ASCII EOT) kann zur Erzeugung eines Dateiendes von einem Terminal verwendet werden. Wird dieses Zeichen empfangen, werden alle Zeichen, die noch zu lesen sind sofort an das Programm übergeben, ohne daß auf ein Neue Zeile-Zeichen gewartet wird, und das EOF-Zeichen wird weggeworfen. Sind keine Zeichen mehr zu lesen da EOF am Anfang einer Zeile steht, werden Null-Zeichen übergeben. Dies ist das Standardzeichen für Dateiende.
NL	(ASCII LF) ist die normale Zeilenbegrenzung. Sie kann weder geändert noch durch ein Escape unwirksam gemacht werden.
EOL	(ASCII NUL) ist ein zusätzliches Zeilenbegrenzungszeichen wie NL. Es wird normalerweise nicht verwendet.
STOP	(Control-s oder ASCII DC3) dient zum vorübergehenden Stoppen der Ausgabe. Es wird häufig gebraucht, um die Bildschirmausgabe anzuhalten. Solange die Ausgabe suspendiert ist, werden STOP-Zeichen ignoriert und nicht gelesen.
START	(Control-q oder ASCII DC1) dient zum Restart der Bildschirmausgabe, die mit einem STOP-Zeichen ausgesetzt wurde. Solange die Ausgabe nicht ausgesetzt ist, werden START-Zeichen ignoriert und nicht gelesen. Die Start/Stop-Zeichen können nicht geändert und nicht mit Escape unwirksam gemacht werden.

Die Zeichenwerte für INTR, QUIT, SWTCH, ERASE, KILL, EOF und EOL können Sie individuell ändern. Die Zeichen ERASE, KILL und EOF können durch Voranstellung des Escape-Zeichens (\) unwirksam gemacht werden.

Anhang 2: Gerätedateien

Fällt das Trägersignal des Modem aus, wird an alle Prozesse, für die dieses Terminal das Steuerterminal ist, ein Hangup-Signal gesendet. Wenn keine anderen Vereinbarungen getroffen sind, bricht dieses Signal die Prozesse ab. Wird das Hangup-Signal ignoriert, endet jedes weitere Lesekommando mit einem Dateiende-Zeichen. Daher können Programme, die von einem Terminal lesen und das Dateiende abfragen, ordnungsgemäß enden, wenn die Modemleitung unterbrochen ist.

Werden ein oder mehr Zeichen geschrieben, werden sie an das Terminal übergeben, sobald die vorher geschriebenen Zeichen ausgegeben sind. Eingabezeichen werden im Echoverfahren ausgegeben, indem sie in die Ausgabewarteschlange gestellt werden, sobald sie eintreffen. Erzeugt ein Prozeß schneller Zeichen als diese ausgegeben werden können, wird der Prozeß suspendiert, sobald seine Ausgabewarteschlange eine gewisse Grenze überschreitet. Wenn die Ausgabewarteschlange wieder bis auf einen gewissen Schwellenwert zusammengeschmolzen ist, wird das Programm fortgesetzt.

Für Terminaldateien gelten mehrere ioctl-Systemaufrufe. Die primären Aufrufe benutzen die folgende, in <termio.h> definierte, Struktur:

```
#define      NCC 8
struct      termio
{
    unsigned  short  ciflag; /* input modes */
    unsigned  short  coflag; /* output modes */
    unsigned  short  ccflag; /* control modes */
    unsigned  short  clflag; /* local modes */
    char      cline; /* line discipline */
    unsigned  char   ccc[NCC]; /* control chars */
};
```

A2

Anhang 2: Gerätedateien

Die speziellen Steuerzeichen werden durch das Array `c_cc` definiert. Die Anfangswerte der einzelnen Funktionen haben die folgenden relativen Positionen:

0	INTR	DEL
1	VQUIT	FS
2	VERASE	#
3	VKILL	@
4	VEOF	EOT
5	VEOL	NUL
6	reserved	
7	SWTCH	

Das Feld `c_iflag` beschreibt die Grundzüge der Eingabesteuerung für das Terminal:

IGNBRK	0000001	Break-Bedingung ignorieren.
BRKINT	0000002	Bei break ein Interrupt signalisieren.
IGNPAR	0000004	Zeichen mit Paritätsfehlern ignorieren.
PARMRK	0000010	Paritätsfehler melden.
INPCK	0000020	Paritätsprüfung für Eingabe aktivieren.
ISTRIP	0000040	Zeichen verkürzen.
INLCR	0000100	NL in CR umwandeln.
IGNCR	0000200	CR ignorieren.
ICRNL	0000400	CR in NL umwandeln.
IUCLC	0001000	Großbuchstaben in Kleinbuchstaben umwandeln.
IXON	0002000	Start/Stop-Ausgabekontrolle aktivieren.
IXANY	0004000	Beliebiges Zeichen für Neustart der Ausgabe aktivieren.
IXOFF	0010000	Start/Stop-Eingabekontrolle aktivieren.

Wenn `IGNBRK` gesetzt ist, wird die Break-Bedingung ignoriert, d. h. es wird nichts in die Eingabewarteschlange eingegeben und daher nichts von irgendeinem Prozeß gelesen. Wenn jedoch `BRKINT` gesetzt ist, erzeugt die Break-Bedingung ein Interrupt-Signal und entleert die Eingabe- ebenso wie die Ausgabewarteschlange. Wenn `IGNPAR` gesetzt ist, werden Zeichen mit anderen Framing- und Paritätsfehlern ignoriert.

Anhang 2: Gerätedateien

Wenn PARMRK gesetzt ist, wird ein Zeichen mit einem Framing- oder Paritätsfehler, das nicht ignoriert wird, als die drei Zeichen lange Sequenz 0377, 0, X gelesen, wobei X die Daten des fehlerhaft empfangenen Zeichens darstellt. Um eine Zweideutigkeit in diesem Fall zu vermeiden, sofern ISTRIP nicht gesetzt ist, wird das gültige Zeichen 0377 als 0377, 0377 gelesen. Wenn PARMRK nicht gesetzt ist, wird ein Framing- oder Paritätsfehler, der nicht ignoriert wird, als das Zeichen NUL (0) gelesen.

Ist INPCK gesetzt, ist die Paritätsprüfung für die Eingabe aktiviert. Wenn INPCK nicht gesetzt ist, ist die Paritätsprüfung für die Eingabe deaktiviert. In diesem Fall kann Ausgabeparität ohne Eingabeparitätsfehler erzeugt werden.

Ist ISTRIP gesetzt, werden gültige Eingabezeichen zunächst auf 7 Bits verkürzt; andernfalls werden alle 8 Bits verarbeitet.

Ist INLCR gesetzt, wird ein empfangenes NL-Zeichen in ein CR-Zeichen umgesetzt. Wenn IGNCR gesetzt ist, wird ein empfangenes CR-Zeichen ignoriert (nicht gelesen). Andernfalls, wenn ICRNL gesetzt ist, wird ein empfangenes CR-Zeichen in ein NL-Zeichen umgesetzt.

Ist IUCLC gesetzt, wird ein empfangener Großbuchstabe in den entsprechenden Kleinbuchstaben umgesetzt.

Ist IXON gesetzt, ist die Ausgabeflußkontrolle im Start/Stop-Verfahren aktiviert. Ein empfangenes STOP-Zeichen stoppt die Ausgabe, und ein empfangenes START-Zeichen startet die Ausgabe wieder. Alle Start/Stop-Zeichen werden ignoriert und nicht gelesen. Wenn IXANY gesetzt ist, bewirkt jedes beliebige Eingabezeichen den Neustart der suspendierten Ausgabeoperation.

Ist IXOFF gesetzt, sendet das System START/STOP-Zeichen aus, wenn die Eingabewarteschlange nahezu leer bzw. voll ist.

Der anfängliche Wert für die Eingabesteuerung ist „alle Bits gelöscht“.

Anhang 2: Gerätedateien

Das Feld `c_oflag` gibt an, wie das System die Ausgabe behandelt:

OPOST	0000001	Ausgabe nachverarbeiten.
OLCUC	0000002	Kleinbuchstaben als Großbuchstaben ausgeben.
ONLCR	0000004	NL bei Ausgabe zu CR-NL machen.
OCRNL	0000010	CR bei Ausgabe zu NL machen.
ONOCR	0000020	Keine Ausgabe von CR in Spalte 0.
ONLRET	0000040	NL hat die Wirkung von CR.
OFILL	0000100	Füllzeichen für Zeitverzögerung verwenden.
OFDEL	0000200	Füllzeichen ist DEL, andernfalls NUL.
NLDLY	0000400	Zeitverzögerungen für Newline festlegen:
NL0	0	
NL1	0000400	
CRDLY	0003000	Zeitverzögerungen für Wagenrücklauf festlegen:
CR0	0	
CR1	0001000	
CR2	0002000	
CR3	0003000	
TABDLY	0014000	Zeitverzögerungen für Tabulator:
TAB0	0	
TAB1	0004000	
TAB2	0010000	
TAB3	0014000	Tabs in Leerzeichen expandieren.
BSDLY	0020000	Zeitverzögerungen für Backspace festlegen:
BS0	0	
BS1	0020000	
VTDLY	0040000	Zeitverzögerungen für Vertikaltabulator festlegen:
VT0	0	
VT1	0040000	
FFDLY	0100000	Zeitverzögerungen für Formularvorschub festlegen:
FF0	0	
FF1	0100000	

Anhang 2: Gerätedateien

Ist OPOST gesetzt, werden die Ausgabebezeichnungen so nachverarbeitet, wie dies von den weiteren Flags angegeben wird; andernfalls werden die Zeichen unverändert übergeben.

Ist OLCUC gesetzt, wird ein Kleinbuchstabe als der entsprechende Großbuchstabe übergeben. Diese Funktion wird häufig in Verbindung mit IUCLC verwendet.

Ist ONLCR gesetzt ist, wird das NL-Zeichen als Zeichenpaar CR-NL übergeben. Wenn OCRNL gesetzt ist, wird das CR-Zeichen als NL-Zeichen übergeben. Ist ONOCR gesetzt, wird kein CR-Zeichen übergeben, wenn man sich in Spalte 0 (erste Position) befindet. Wenn ONLRET gesetzt ist, wird davon ausgegangen, daß das NL-Zeichen die Funktion eines Wagenrücklaufs hat; der Spaltenzeiger wird auf 0 gesetzt und die für CR spezifizierten Zeitverzögerungen angewandt. Andernfalls wird davon ausgegangen, daß das NL-Zeichen nur die Zeilenvorschubfunktion hat; der Spaltenzeiger bleibt dann unverändert. Der Spaltenzeiger wird auch dann auf 0 gesetzt, wenn effektiv das CR-Zeichen übergeben wird.

Die Verzögerungsbits geben an, wie lange die Übergabe von Zeichen gestoppt wird, damit mechanische oder sonstige Bewegungen ausgeführt werden, nachdem bestimmte Zeichen an das Terminal übergeben wurden. Der Wert 0 bedeutet in jedem Fall keine Verzögerung. Wenn OFILL gesetzt ist, werden für die Zeitverzögerung Füllzeichen statt einer zeitlich festgelegten Verzögerung übergeben. Dies ist sinnvoll bei Terminals mit einer hohen Baudrate, für die nur eine sehr geringe Verzögerung erforderlich ist. Wenn OFDEL gesetzt ist, ist das Füllzeichen DEL, sonst NUL.

Ist eine Zeitverzögerung für einen Formularvorschub oder Vertikaltabulator angegeben, dauert diese ca. 2 Sekunden.

Die Zeitverzögerung für Newline dauert ca. 0,10 Sekunden. Wenn ONLRET gesetzt ist, werden die Zeitverzögerungen für Wagenrücklauf statt derjenigen für Newline verwendet. Wenn OFILL gesetzt ist, werden zwei Füllzeichen übergeben.

© ...Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Mitteilung ihres Inhalts nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zu Schadenersatz. Alle Rechte für den Fall der Patenterteilung oder Gebrauchsmustererteilung vorbehalten.

Anhang 2: Gerätedateien

Die Wagenrücklaufverzögerung Typ 1 hängt von der aktuellen Spaltenposition ab, Typ 2 dauert ca. 0,10 Sekunden und Typ 3 ca. 0,15 Sekunden. Wenn OFILL gesetzt ist, werden bei Verzögerung Typ 1 zwei Füllzeichen und bei Verzögerung Typ 2 vier Füllzeichen übergeben.

Die Horizontaltabulator-Verzögerung Typ 1 hängt von der aktuellen Spaltenposition ab. Typ 2 dauert ca. 0,10 Sekunden. Typ 3 gibt an, daß Tabs zu Leerzeichen expandiert werden sollen. Wenn OFILL gesetzt ist, werden für jede Verzögerung zwei Füllzeichen übergeben.

Die Backspace-Verzögerung dauert ca. 0,05 Sekunden. Wenn OFILL gesetzt ist, wird ein Füllzeichen übergeben.

Die effektiven Zeitverzögerungen hängen von der Leitungsgeschwindigkeit und der Systembelastung ab.

Der anfängliche Wert für die Ausgabesteuerung ist „alle Bits gelöscht“.

Anhang 2: Gerätedateien

Das Feld c_cflag beschreibt die hardwaremäßige Steuerung des Terminals:

CBAUD	000017	Baudrate:
B0	0	Hangup
B50	000001	50 Baud
B75	000002	75 Baud
B110	000003	110 Baud
B134	000004	134,5 Baud
B150	000005	150 Baud
B200	000006	200 Baud
B300	000007	300 Baud
B600	000010	600 Baud
B1200	000011	1200 Baud
B1800	000012	1800 Baud
B2400	000013	2400 Baud
B4800	000014	4800 Baud
B9600	000015	9600 Baud
EXTA	000016	External A
EXTB	000017	External B
CSIZE	000060	Zeichenlänge:
CS5	0	5 Bits
CS6	000020	6 Bits
CS7	000040	7 Bits
CS8	000060	8 Bits
CSTOPB	0000100	Zwei Stoppbits senden, andernfalls eins.
CREAD	0000200	Empfänger aktivieren.
PARENB	0000400	Parität aktivieren.
PARODD	0001000	Ungerade Parität, sonst gerade.
HUPCL	0002000	Hangup nach letztem close.
CLOCAL	0004000	Lokale Leitung, andernfalls Wählleitung.
LOBLK	0010000	Layer-Ausgabe geblockt.

Die CBAUD-Bits geben die Baudrate an. Mit der Baudrate Null (B0) wird die Verbindung getrennt. Wenn B0 angegeben ist, wird das Signal Data Terminal Ready (DTR) nicht als aktiv gemeldet. Dadurch wird die Leitung normalerweise getrennt. Für eine bestimmte Hardware unmögliche Änderungen der Übertragungsgeschwindigkeit werden ignoriert.

Anhang 2: Gerätedateien

Die CSIZE-Bits geben die Zeichenlänge in Bits für Sendung und Empfang an. In der Zeichenlänge ist ein ggf. vorhandenes Paritätsbit nicht enthalten. Wenn CSTOPB gesetzt ist, werden zwei Stoppbits verwendet, andernfalls ein Stoppbit. Bei 110 Baud sind beispielsweise zwei Stoppbits erforderlich.

Ist PARENB gesetzt, wird die Paritätserzeugung und -erkennung aktiviert, und jedes Zeichen wird um ein Paritätsbit ergänzt. Wenn Parität aktiviert ist, wird durch das gesetzte PARODD-Flag ungerade Parität angegeben, während andernfalls mit gerader Parität gearbeitet wird.

Wenn CREAD gesetzt ist, ist der Empfänger aktiviert. Andernfalls werden keine Zeichen empfangen.

Ist HUPCL gesetzt, wird die Leitung getrennt, sobald der letzte Prozeß, der die Leitung offenhält, diese schließt oder selbst endet. Das heißt, daß das Signal Data Terminal Ready (DTR) nicht als aktiv gemeldet wird.

Wenn CLOCAL gesetzt ist, handelt es sich um eine lokale Leitung mit direkter Verbindung ohne Modemsteuerung. Andernfalls wird von Modemsteuerung ausgegangen.

Ist LOBLK gesetzt, wird die Ausgabe des Job Control Layers geblockt, solange es sich nicht um den aktuellen Layer handelt. Andernfalls wird die Ausgabe im Multiplex-Verfahren zum aktuellen Layer übertragen.

Der anfängliche hardwaremäßige Wert nach dem Eröffnen ist B300, CS8, CREAD, HUPCL.

Anhang 2: Gerätedateien

Das Feld `c_iflag` der Argumentstruktur wird von der Leitungsprozedur zur Steuerung der Terminalfunktionen verwendet. Das grundlegende Leitungsprotokoll (0) sieht folgendes vor:

ISIG	0000001	Signale aktivieren.
ICANON	0000002	Verarbeitung von <code>erase</code> und <code>kill</code> .
XCASE	0000004	Darstellung von Groß-/Kleinbuchstaben.
ECHO	0000010	Echo aktivieren.
ECHOE	0000020	Zeilenlöschzeichen als BS-SP-BS zurückmelden.
ECHOK	0000040	NL nach Zeilenlöschzeichen zurückmelden.
ECHONL	0000100	NL zurückmelden.
NOFLSH	0000200	Kein Flush nach Interrupt oder Quit.

Wenn ISIG gesetzt ist, wird jedes Eingabezeichen mit den speziellen Steuerzeichen INTR und QUIT verglichen. Entspricht ein Eingabezeichen einem dieser Steuerzeichen, wird die dem Steuerzeichen zugeordnete Funktion ausgeführt. Ist ISIG nicht gesetzt, erfolgt keine entsprechende Prüfung. Diese speziellen Eingabefunktionen sind also nur möglich, wenn ISIG gesetzt ist. Diese Funktionen können einzeln deaktiviert werden, indem Sie den Wert des Steuerzeichens auf einen unwahrscheinlichen oder unmöglichen Wert (z. B. 0377) setzen.

Ist ICANON gesetzt, ist die normale Verarbeitung aktiviert. Dadurch werden die Funktionen für Zeichenlöschung (`erase`) und Zeilenlöschung (`kill`) aktiviert, und die Eingabezeichen werden in Form von Zeilen, die durch NL, EOF und EOL abgeschlossen werden, eingegeben. Ist ICANON nicht gesetzt, werden Leseanforderungen direkt aus der Eingabewarteschlange erfüllt. Ein Lesewunsch wird erst dann erfüllt, wenn entweder MIN Zeichen empfangen wurden oder der Timeout-Wert TIME zwischen den Zeichen abgelaufen ist. Dadurch werden einerseits Zeichengruppen bei der Eingabe zusammengefaßt und effizient gelesen, während andererseits aber auch einzelne Zeichen eingegeben werden können. Die Werte für MIN und TIME werden auf der Position für die Zeichen EOF bzw. EOL abgelegt. Der TIME-Wert wird in Zehntelsekunden angegeben.

Anhang 2: Gerätedateien

Wenn XCASE und ICANON beide gesetzt sind, wird ein Großbuchstabe bei der Eingabe übernommen, indem ihm ein Backslash \ vorangestellt wird, und bei der Ausgabe wird ihm ebenfalls ein Backslash \ vorangestellt. In diesem Modus werden die folgenden Escape-Folgen bei der Ausgabe erzeugt und bei der Eingabe akzeptiert:

Für:	Verwenden Sie:
'	\'
	\
~	\~
{	\{
}	\}
\	\\

BEISPIELE:

A wird als \a, \n wird als \\n und \N als \\N eingegeben.

Wenn ECHO gesetzt ist, werden die empfangenen Zeichen angezeigt.

Ist ICANON gesetzt, sind die folgenden Echofunktionen möglich. Wenn ECHO und ECHOE gesetzt sind, wird das Zeichenlöschzeichen (erase) als ASCII BS SP BS zurückgemeldet und löscht das letzte Zeichen vom Bildschirm. Wenn ECHOE gesetzt ist und ECHO nicht gesetzt ist, wird das Zeichenlöschzeichen (erase) als ASCII SP BS zurückgemeldet. Ist ECHOK gesetzt, wird das NL-Zeichen nach dem Zeilenlöschzeichen zurückgemeldet, um nachdrücklich darauf hinzuweisen, daß die Zeile gelöscht wird. Wird vor dem Zeichen für Zeichenlöschung oder Zeilenlöschung ein Escape-Zeichen angegeben, so wird dadurch die Sonderfunktion aufgehoben. Wenn ECHONL gesetzt ist, wird das NL-Zeichen zurückgemeldet, auch wenn ECHO nicht gesetzt ist. Dies ist sinnvoll bei Terminals, die auf lokales Echo (sogenannte Halbduplex-Terminals) eingestellt sind. Das EOF-Zeichen wird nur dann angezeigt, wenn ihm ein Escape vorangestellt ist. Da EOT das standardmäßige EOF-Zeichen ist, wird dadurch verhindert, daß Terminals, die auf EOT reagieren, die Verbindung trennen.

Anhang 2: Gerätedateien

Ist NOFLSH gesetzt, erfolgt die normale Entleerung der Eingabe- und Ausgabewarteschlange bei den Zeichen für Quit, Switch und Interrupt nicht.

Der anfängliche Kontrollwert für die Leitungsprozedur ist „alle Bits gelöscht“.

Die primären ioctl-Systemaufrufe haben folgende Form:

```
ioctl (Dateibescheidungsnummer, Kommando, Argument)
struct termio *arg;
```

Folgende Kommandos arbeiten in dieser Form:

- | | |
|---------|--|
| TCGETA | Die zu dem Terminal gehörenden Parameter holen und in der von Argument angegebenen termio-Struktur ablegen. |
| TCSETA | Die zu dem Terminal gehörenden Parameter aufgrund der vom Argument angesprochenen Struktur setzen. Die Änderung wirkt sofort. |
| TCSETAW | Warten, bis die Ausgabewarteschlange leer ist, bevor die neuen Parameter gesetzt werden. Diese Form sollte verwendet werden, wenn Parameter geändert werden, die die Ausgabe beeinflussen. |
| TCSETAF | Warten, bis die Ausgabewarteschlange leer ist. Anschließend die Eingabewarteschlange entleeren und die neuen Parameter setzen. |

Weitere ioctl-Aufrufe haben folgende Form:

```
ioctl (Dateibescheidungsnummer, Kommando, Argument)
int arg;
```

A2

Anhang 2: Gerätedateien

Folgende Kommandos arbeiten in dieser Form:

- | | |
|--------|---|
| TCSBRK | Warten, bis die Ausgabewarteschlange leer ist. Wenn Argument gleich 0 ist, ein Break senden (d. h. Nullbits für die Dauer von 0,25 Sekunden). |
| TCXONC | Start/Stop-Steuerung. Ist das Argument gleich 0, Ausgabe suspendieren; ist das Argument gleich 1, suspendierte Ausgabe neu starten. |
| TCFLSH | Ist das Argument gleich 0, die Eingabewarteschlange entleeren; ist das Argument gleich 1, die Ausgabewarteschlange entleeren; ist das Argument gleich 2, die Eingabe- und die Ausgabewarteschlange entleeren. |

DATEIEN:

/dev/tty*

SIEHE AUCH:

Benutzerkommando: stty

Systemschnittstelle: ioctl

Anhang 2: Gerätedateien

tty – Schnittstelle für Steuerterminal

BESCHREIBUNG:

Die Datei /dev/tty ist in allen Prozessen ein Synonym für das Steuerterminal, das der Prozeßgruppe des betreffenden Prozesses, falls vorhanden, zugeordnet ist. Sie ist nützlich für Programme oder Shell-Sequenzen, die Meldungen unbedingt auf dem Terminal ausgeben wollen, gleichgültig wie die Ausgabe umgelenkt wurde. Sie kann auch für Programme verwendet werden, die den Namen einer Datei für die Ausgabe verlangen, wenn Druckausgabe gewünscht wird und es aufwendig ist festzustellen, welches Terminal derzeit belegt ist.

DATEIEN:

/dev/tty

/dev/tty*

Anhang 3: Dateiformate

A3 Dateiformate

acct – Format für Prozeßabrechnungsdateien

SYNTAX:

```
#include <sys/acct.h>
```

BESCHREIBUNG:

Accounting-Dateien, die von acct erstellt werden, beinhalten Sätze in der von <sys/acct.h> definierten Form mit folgendem Inhalt:

```
typedef ushort comp_t; /* "floating point" */
                        /* 13-bit fraction, 3-bit exponent */

struct acct
{
    char  ac_flag;      /* Accounting flag */
    char  ac_stat;     /* Exit status */
    ushort ac_uid;
    ushort ac_gid;
    dev_t ac_tty;
    time_t ac_btime;   /* Beginning time */
    comp_t ac_untime;  /* acctng user time in clock ticks */
    comp_t ac_sstime;  /* acctng system time in clock ticks */
    comp_t ac_etime;   /* acctng elapsed time in clock ticks */
    comp_t ac_mem;     /* memory usage in clicks */
    comp_t ac_io;      /* chars trnsfrd by read/write */
    comp_t ac_rw;      /* number of block reads/writes */
    char  ac_comm[8];  /* command name */
};

extern struct acct      acctbuf;
extern struct inode    *acctp; /* inode of accounting file */

#define AFORK 01      /* has executed fork, but no exec */
#define ASU   02      /* used super-user privileges */
#define ACCTF 0300    /* record type: 00 = acct */
```

acct

Anhang 3: Dateiformate

In `ac_flag` wird das AFORK-Flag von jedem fork gesetzt und von jedem exec gelöscht. Das Feld `ac_comm` wird vom Vaterprozeß geerbt und wird von jedem exec zurückgesetzt. Jedesmal wenn das System den Prozeß mit einem Uhrtakt belastet, addiert es die aktuelle Prozeßgröße zu `ac_mem`, die wie folgt berechnet wird:

$(\text{Datenlänge}) + (\text{Textlänge}) / (\text{Anzahl der im Hauptspeicher befindlichen und textbenutzenden Prozesse})$

Der Wert von $\text{ac_mem} / (\text{ac_stime} + \text{ac_utime})$ kann als Annäherung der mittleren Prozeßgröße, die durch gemeinsame Benutzung von Text modifiziert wird, angesehen werden.

Die Struktur `tacct.h`, die bei den Quelldateien der Abrechnungskommandos liegt, stellt das von den verschiedenen Abrechnungskommandos verwendete Summenabrechnungsformat dar:

```
/*
 * total accounting (for acct period), also for day
 */

struct tacct
{
    uid_t      ta_uid;      /* userid */
    char       ta_name[8];  /* login name */
    float      ta_cpu[2];   /* cum. cpu time, p/np (mins) */
    float      ta_kcore[2]; /* cum kcore-minutes, p/np */
    float      ta_con[2];   /* cum. connect time, p/np, mins */
    float      ta_du;       /* cum. disk usage */
    long       ta_pc;       /* count of processes */
    unsigned short ta_sc;   /* count of login sessions */
    unsigned short ta_dc;   /* count of disk samples */
    unsigned short ta_fee;  /* fee for special services */
};
```

Anhang 3: Dateiformate

HINWEIS:

Der ac_mem-Wert für ein kurzlebiges Kommando gibt nur einen ungenauen Hinweis auf die tatsächliche Größe des Kommandos, weil ac_mem möglicherweise erhöht wird, während ein anderes Kommando (z. B. die Shell) vom Prozeß ausgeführt wird.

SIEHE AUCH:

Administrator-Kommando: acct
Benutzerkommando: acctcom
Systemschnittstellen: acct, exec, fork

Anhang 3: Dateiformate

checklist – Liste der Dateisysteme, die von fsck überprüft werden

BESCHREIBUNG:

Die Datei checklist liegt im Verzeichnis /etc und enthält eine Liste von Gerätedateien (maximal 15). In jeder Zeile darf nur ein Name einer Gerätedatei stehen. Die einzelnen Gerätedateien sind jeweils mit einem Dateisystem verbunden und werden von fsck automatisch überprüft.

SIEHE AUCH:

Administrator-Kommando: fsck

Anhang 3: Dateiformate

core – Format der Core-Image Datei

BESCHREIBUNG:

Das UNIX-Betriebssystem schreibt das Speicherabbild (Core-Image) eines terminierten Prozesses dann, wenn bestimmte Fehler auftreten. Bei der Beschreibung der Systemschnittstelle signal sind die einzelnen Gründe aufgeführt, die zu einer Aufzeichnung des Speicherabblids führen. Die häufigsten Gründe sind illegale Befehle, Bus-Fehler und von Benutzern generierte Quit-Signale. Das Speicherabbild wird in die Datei core geschrieben, die im aktuellen Verzeichnis des betroffenen Prozesses kreiert wird. Ist bei einem Prozeß die effektive Benutzer-ID unterschiedlich zu der realen Benutzer-ID wird kein Core-Image gemacht.

Im ersten Abschnitt des Speicherabblids wird eine Kopie der Systemdaten des Prozesses – einschließlich der Register zum Zeitpunkt des Fehlers – abgelegt. Die Größe dieses Abschnitts ist abhängig vom Parameter `usize`, der in `/usr/include/sys/param.h` definiert ist. Der Rest von core enthält den aktuellen Speicherplatzinhalt des Speicherabschnitts, den der Benutzer belegt hatte. Kann das Textsegment nur gelesen werden, ist es `shared` oder vom Datenteil getrennt, ist es im Speicherabbild nicht enthalten.

Das Format der Informationen im ersten Abschnitt wird in der Datei `/usr/include/sys/user.h` beschrieben. Die Registerplätze finden Sie in `/usr/include/sys/reg.h`.

SIEHE AUCH:

Administrator-Kommando: `crash`
Benutzerkommando: `sdb`
Systemschnittstellen: `setuid`, `signal`

Anhang 3: Dateiformate

cpio – Format des cpio-Archivs

BESCHREIBUNG:

Wird die Option `-c` von `cpio` nicht verwendet, hat der Header die folgende Struktur:

```
struct
{
    short    h_magic,
            h_dev;
    ushort  h_ino,
            h_mode,
            h_uid,
            h_gid;
    short    h_nlink,
            h_rdev,
            h_mtime [2],
            h_namesize,
            h_filesize [2];
    char     h_name [h_namesize rounded to word];
} Hdr;
```

Wenn die Option `-c` angegeben ist, wird die Header-Information wie folgt beschrieben:

```
sscanf(Chdr,"%6o%6o%6o%6o%6o%6o%6o%6o%11lo%6o%11lo%s",
        &Hdr.h_magic, &Hdr.h_dev, &Hdr.h_ino, &Hdr.h_mode,
        &Hdr.h_uid, &Hdr.h_gid, &Hdr.h_nlink, &Hdr.h_rdev,
        &Longtime, &Hdr.h_namesize,&Longfile,Hdr.h_name);
```

`Longtime` und `Longfile` sind gleichbedeutend mit `Hdr.h_mtime` bzw. `Hdr.h_filesize`. Der Inhalt jeder Datei wird in einem Element des Array unterschiedlich langer Strukturen (Archiv), zusammen mit anderen Angaben zur Beschreibung der Datei gespeichert. Jedes Exemplar von `h_magic` enthält die Konstante 070707 (oktal). Die Bedeutung der Angaben `h_dev` bis `h_mtime` ist bei der Systemschnittstelle stat erläutert. Die Länge des durch ein Nullzeichen beendeten Pfadnamens `h_name` einschließlich des Nullzeichens wird von `h_namesize` angegeben.

Anhang 3: Dateiformate

Der letzte Satz des Archivs enthält immer den Namen TRAILER!!!. Gerätedateien, Verzeichnisse und der Nachspann (Trailer) werden mit einer h_filesize von Null verzeichnet.

SIEHE AUCH:

Benutzerkommandos: cpio, find
Systemschnittstelle: stat

Anhang 3: Dateiformate

errfile – Format für Fehlerprotokolldateien

BESCHREIBUNG:

Erkennt das System Hardwarefehler, wird ein Fehlersatz erzeugt und an den Fehlerprotokollierungs-Dämon übergeben, der ihn für die spätere Auswertung im Fehlerprotokoll aufzeichnet. Das standardmäßige Fehlerprotokoll ist /usr/adm/errfile.

Das Format eines Fehlersatzes hängt von der Art des Fehlers ab. Jeder Fehlersatz besitzt jedoch einen Header mit folgendem Format:

```
struct errhdr
{
    short      e_type;      /* record type */
    short      e_len;      /* bytes in record + hdr) */
    time_t     e_time;     /* time of day */
    int        e_cpu;      /* proc recording error - 3B 20A
                           Computer only */
};
```

Folgende Satzarten sind zulässig:

```
#define E_GOTS 010      /* start for UNIX System Release 3.0*/
#define E_GORT 011      /* start for UNIX system/RT */
#define E_STOP 012      /* stop */
#define E_TCHG 013      /* time change */
#define E_CCHG 014      /* configuration change */
#define E_BLK 020       /* block device error */
#define E_STRAY 030     /* stray interrupt */
#define E_PRTY 031     /* memory parity */
#define E_PIO 041       /* 3B 20S computer programmed I/O */
#define E_IOP 042       /* 3B 20S computer I/O processor */
```

Einige Sätze in der Fehlerdatei haben administrativen Charakter. Dazu zählt der Anfangssatz, der bei Aktivierung der Protokollierung in die Datei eingetragen wird, der Endesatz, der in die Datei geschrieben wird, wenn der Dämon geregelt beendet wird, und der Zeitwechselsatz, der dafür sorgt, daß ein Wechsel in der Systemzeit berücksichtigt wird. Diese Sätze haben folgende Formate:

Anhang 3: Dateiformate

```

struct estart
{
    short          e_cpu;          /* CPU type */
    struct utsname e_name;        /* system names */
#ifdef u3b
    short          e_mmr3;        /* contents mem mgmt reg 3 */
    long           e_sysssize;    /* 11/70 system memory size */
    short          e_bconf;      /* block dev configuration */
#endif
#ifdef u3b
    int            e_mmcnt;       /* kbytes per array */
#endif
};

#define eend errhdr /* record header */

struct etimchg
{
    time_t         e_ntime;       /* new time */
};
    
```

Stray-Interrupts bewirken Sätze mit folgendem Format:

```

struct estray
{
#ifdef u3b
    uint          e_saddr;        /* stray loc or device addr */
#else
    physadr       e_saddr;        /* stray loc or device addr */
    short         e_sbacty;       /* active block devices */
#endif
};
    
```

Anhang 3: Dateiformate

Speicher-Subsystemfehler auf 3B 20 Rechnern und VAX 11/70 Prozessoren bewirken Sätze mit folgendem Format:

```

struct eparity
{
#ifdef u3b
    int            e_parreg[3]; /* 3B computer memory
                               registers */
#else
    short         e_parreg[4]; /* memory subsys registers */
#endif
};

```

Speicher-Subsystemfehler auf VAX-11/780 Prozessoren bewirken Sätze mit folgendem Format:

```

struct ememory
{
    int            e_sbier;
    int            e_memcad;
};

```

Fehlersätze für blockorientierte Geräte haben das folgende Format:

```

struct eblock
{
#ifdef u3b
    ushort        e_num;      /* device number */
    struct iostat
    {
        long      io_ops;     /* number read/writes */
        long      io_misc;    /* number "other" operations */
        ushort    io_unlog;   /* number unlogged errors */
    }             e_stats;
    short         e_bflags;   /* read/write, error, etc */
    daddr_t       e_bnum;     /* logical block number */
    uint          e_bytes;    /* number bytes to transfer */
    union ptbl
    {
        int page[64];        /* page table entries */
        union ptbl *pnext;
    }             e_ptbl;
    struct ptbl   e_ptbl;    /* page table for transfer */
#endif
};

```

Anhang 3: Dateiformate

```

uint          e_yoff;      /* offset into page table */
uint          e_stat1;    /* status word 1 */
uint          e_stat2;    /* status word 2 */
#endif
#ifdef u3b
dev_t         e_dev;      /* "true" major + minor dev no */
physadr       e_regloc;   /* controller address */
short         e_bacty;    /* other block I/O activity */
struct iostat
{
    long       io_ops;     /* number read/writes */
    long       io_misc;    /* number "other" operations */
    ushort     io_unlog;   /* number unlogged errors */
    e_stats;
}
short         e_bflags;   /* read/write, error, etc */
short         e_cyloff;   /* logical dev start cyl */
daddr_t       e_bnum;     /* logical block number */
ushort        e_bytes;    /* number bytes to transfer */
paddr_t       e_memadd;   /* buffer memory address */
ushort        e_rtry;     /* number retries */
short         e_preg;     /* number device registers */
#endif
#ifdef vax
struct mba_regs
{
    long mba_csr;
    long mba_cr;
    long mba_sr;
    long mba_var;
    long mba_vcr;
} e_mba;
#endif
};

```

Folgende Werte werden in e_bflags benutzt:

```

#define E_WRITE      0      /* write operation */
#define E_READ       1      /* read operation */
#define E_NOIO       02     /* no I/O pending */
#define E_PHYS       04     /* physical I/O */
#define E_MAP        010    /* Unibus map in use */
#define E_ERROR      020    /* I/O failed */

```

Anhang 3: Dateiformate

Folgende Fehlersätze sind nur für 3B 20 Rechner:

```

struct epio                                /* programmed I/O (pio) error */
{
    char        e_chan;                    /* which channel */
    char        e_dev;                    /* which dev on channel */
    uint        e_chstat;                 /* channel status */
    uint        e_cmd;                    /* pio command */
}

struct iop                                  /* I/O processor (iop) error */
{
    char        e_unit;                   /* unit number */
    uint        e_word0;                  /* iop report word */
    uint        e_word1;                  /* iop report word */
}

```

Die „wahren“ Major-Nummern um Geräte zu identifizieren sind die folgenden:

Digital Equipment	Western Electric
#define RK0 0	#define DFC0 0
#define RP0 1	#define IOP0 1
#define RF0 2	#define MT0 2
#define TM0 3	
#define TC0 4	
#define HP0 5	
#define HT0 6	
#define HS0 7	
#define RL0 8	
#define HP1 9	
#define HP2 10	
#define HP3 11	

SIEHE AUCH:

Administrator-Kommando: errdemon

Anhang 3: Dateiformate

fs, inode – Format eines Datenträgers für Dateisysteme

SYNTAX:

```
#include <sys/filsys.h>
#include <sys/types.h>
#include <sys/param.h>
```

BESCHREIBUNG:

Alle Datenträger für Dateisysteme besitzen für bestimmte wichtige Informationen ein gemeinsames Format. Jeder dieser Datenträger ist in eine bestimmte Anzahl Blöcke unterteilt. Block 0 ist leer und steht zur Verfügung, um ein Bootstrap-Programm oder andere Informationen aufzunehmen.

Block 1 ist der sogenannte Superblock. Der Superblock hat folgendes Format:

```
/*
 * Structure of the super-block
 */
struct filsys
{
    ushort s_ysize;           /* size in blocks of i-list */
    daddr_t s_fsize;         /* size in blocks of entire volume */
    short s_nfree;           /* number of addresses in sfree */
    daddr_t s_free[NICFREE]; /* free block list */
    short s_ninode;          /* number of i-nodes in sinode */
    ino_t s_inode[NICINOD]; /* free i-node list */
    char s_flock;            /* lock during free list manipulation */
    char s_iloc;             /* lock during i-list manipulation */
    char s_fmod;             /* super block modified flag */
    char s_ronly;           /* mounted read-only flag */
    time_t s_time;          /* last super block update */
    short s_dinfo[4];        /* device information */
    daddr_t s_tfree;         /* total free blocks*/
    ino_t s_tinode;          /* total free i-nodes */
    char s_fname[6];         /* file system name */
    char s_fpack[6];         /* file system pack name */
};
```

Anhang 3: Dateiformate

```

    long    s_fill[13];    /* ADJUST to make sizeof filsys be 512 */
    long    s_magic;      /* magic number to denote new file
                           system */
    long    s_type;       /* type of new file system */
};

#define FsMAGIC 0xfd187e20 /* s_magic number */

#define Fs1b 1           /* 512 byte block */
#define Fs2b 2           /* 1024 byte block */
#define FS4b 4           /* 2048 byte block */

```

S_type indiziert den Typ des Dateisystems. Unterstützt wird zur Zeit eine Blocklänge von 2048 Bytes.

S_magic dient dazu, die Arten der Dateisysteme zu unterscheiden. Ist dieses Feld nicht gleich FsMAGIC, wird Typ Fs1b interpretiert, andernfalls wird das Feld s_type benutzt. In der folgenden Beschreibung wird dann ein Block durch den Typ bestimmt. Für ein 512 Bytes orientiertes Dateisystem beträgt die Blocklänge 512 Bytes. In einem 1024 Bytes orientierten Dateisystem beträgt die Blocklänge 1024 Bytes oder zwei Sektoren. Das Betriebssystem ist verantwortlich für alle Umwandlungen logischer Blocknummern in physikalische Sektornummern.

S_1size ist die Adresse des ersten Datenblocks nach der I-Liste. Die I-Liste beginnt direkt hinter dem Superblock, also in Block 2. Die I-Liste ist also immer s_size - 2 Blöcke groß.

S_fsiz ist der erste Block, der für die Zuordnung von Dateien nicht mehr zur Verfügung steht. Diese Blocknummern werden vom System zum „Bad Block Check“ benutzt. Wird eine „unmögliche“ Blocknummer von der freien Liste belegt, wird eine Meldung auf der Konsole ausgegeben. Außerdem wird der freie Bereich bereinigt.

Die Liste der freien Blöcke (free list) wird nach einem bestimmten Schema instandgehalten. Der s_free-Bereich enthält in s_free[1], ..., s_free[s_nfree-1] bis zu 49 Nummern von freien Blöcken. S-free[0] ist die Blocknummer des Anfangs der Kette, die die freie Liste bildet. Die erste Nummer in jedem Block dieser Kette ist die Anzahl (bis zu 50) der Blocknummern, die in den nächsten 50 Blöcken dieses Kettenglie-

Anhang 3: Dateiformate

des aufgelistet sind. Der erste dieser 50 Blöcke enthält den Verweis zum nächsten Glied der Kette. Um einen Block zuzuordnen, wird `s_nfree` dekrementell vermindert und der neue Block ist `s_free[s_nfree]`. Ist die neue Blocknummer 0, führt dies zu einem Fehler. Wird `s_nfree` 0, wird der Block, der von der neuen Blocknummer referiert wird, gelesen. `S_nfree` wird durch das erste Wort ersetzt und die Blocknummern werden in den `s_free`-Bereich kopiert. Um die Zuordnung eines Blocks zu entfernen, wird überprüft, ob `s_nfree` gleich 50 ist. Ist dies der Fall, werden `s_nfree` und der `s_free`-Bereich kopiert, geschrieben und `s_nfree` auf 0 gesetzt. In jedem Fall wird `s_free[s_nfree]` die Nummer des entfernten Blocks zugewiesen und `s_nfree` wird inkrementiert.

`S_tfree` enthält die Anzahl der freien Blöcke, die im Dateisystem vorhanden sind.

`S_ninode` ist die Anzahl der freien I-Nummern im `s_inode`-Bereich. Um einen I-Knoten zuzuordnen, wird folgendermaßen vorgegangen: Ist `s_ninode` größer als 0, wird es dekrementiert und `s_inode[s_ninode]` zurückgeliefert. Ist `s_ninode` gleich 0, wird die I-Liste gelesen und die Nummern aller freien I-Knoten (bis zu 100) werden im `s_inode`-Bereich plaziert. Anschließend wird der Versuch neu gestartet. Um einen I-Knoten freizugeben, muß `s_ninode` kleiner als 100 sein. Die Nummer des I-Knotens wird in `s_inode[s_ninode]` plaziert und `s_ninode` wird dekrementiert. Ist `s_ninode` bereits auf 100, wird der freie I-Knoten in eine andere Tabelle eingetragen. Die Liste der I-Knoten dient lediglich dazu, den Allokierungsprozeß zu beschleunigen. Die Information, ob ein I-Knoten wirklich frei ist oder nicht, befindet sich im I-Knoten selbst.

`S_tinode` ist die Summe der freien I-Knoten im Dateisystem.

`S_flock` und `s_iloc` sind Flags, die in der Speicherkopie des Dateisystems gehalten werden, solange das Dateisystem eingehängt ist. Die Werte dieser Flags auf der Platte sind nicht bedeutsam. `S-fmod` wird benutzt um anzuzeigen, daß sich der Superblock geändert hat und während der nächsten periodischen Aktualisierung der Dateisysteminformationen auf die Platte kopiert werden muß.

`S_ronly` indiziert einen Schreibschutz (read-only).

Anhang 3: Dateiformate

S_time enthält den letzten Zeitpunkt einer Änderung des Superblocks. Es ist die Anzahl von Sekunden, die seit dem 1.1.1970 0 Uhr verstrichen sind. Während des Bootens wird s_time des Root-Dateisystems dazu benutzt, um die Systemzeit zu setzen.

S_fname enthält den Namen des Dateisystems und s_fpack den Namen des Packs.

I-Nummern beginnen mit 1 und die Speicherung der I-Knoten beginnt in Block 2. I-Knoten sind 64 Bytes lang. I-Knoten 1 ist für eine spätere Benutzung reserviert. I-Knoten 2 enthält das Wurzelverzeichnis des Dateisystems. Andere I-Nummern haben keine Sonderbedeutung. Jeder I-Knoten repräsentiert eine Datei. Eine Beschreibung des I-Knoten-Formats finden Sie unter inode.

DATEIEN:

/usr/include/sys/filsys.h

/usr/include/sys/stat.h

SIEHE AUCH:

Administrator-Kommandos: fsck, fsdb, mkfs

Dateiformat: inode

Anhang 3: Dateiformate

gettydefs – Von getty verwendete Übertragungsgeschwindigkeit und Terminaleinstellungen

BESCHREIBUNG:

Die Datei /etc/gettydefs enthält Informationen, mit deren Hilfe getty die Übertragungsgeschwindigkeit und die Terminaleinstellungen für eine Leitung festlegt. Sie gibt an, wie der Login-Prompt aussehen soll. Ferner enthält sie die Übertragungsgeschwindigkeit, die als nächste versucht werden soll, wenn der Benutzer durch Eingabe eines <break>-Zeichens anzeigt, daß die aktuelle Geschwindigkeit nicht brauchbar ist.

Alle Einträge in /etc/gettydefs haben das folgende Format:

label # initial-flags # final-flags # login-prompt # next-label

Auf jeden Eintrag folgt eine Leerzeile. Die verschiedenen Felder können Sonderzeichen in der Form \b, \n, \c usw. sowie \nnn enthalten, wobei nnn der Oktalwert des gewünschten Zeichens ist. Die Felder haben folgende Bedeutung:

label Zeichenkette, mit der getty sein zweites Argument abzugleichen versucht. Dies ist häufig die Geschwindigkeit, z. B. 1200, mit der das Terminal arbeiten soll, es kann sich aber auch um einen anderen Wert handeln (siehe unten).

initial-flags Diese Flags sind die anfänglichen Werte der Systemchnittstelle ioctl, auf die das Terminal gesetzt werden soll, wenn für getty kein Terminaltyp angegeben wird. Getty versteht diejenigen Flags, die in /usr/include/sys/termio.h aufgeführt sind. Normalerweise muß bei den initial-flags nur das Flag für die Übertragungsgeschwindigkeit angegeben werden. Getty stellt das Terminal automatisch auf raw-Eingabemodus ein und sorgt für die korrekte Einstellung der meisten anderen Flags. Die Einstellungen der initial-flags bleiben wirksam, bis getty login ausführt.

Anhang 3: Dateiformate

<i>final-flags</i>	Die final-flags nehmen dieselben Werte an wie die initial-flags und werden unmittelbar bevor getty login ausgeführt gesetzt. Das Flag für die Übertragungsgeschwindigkeit ist auch hier wieder zwingend erforderlich. Das zusammengesetzte Flag SANE sorgt dafür, daß die meisten anderen Flags korrekt gesetzt werden, so daß der Prozessor und das Terminal auf vernünftige Weise miteinander kommunizieren. Die beiden anderen häufig spezifizierten final-flags sind TAB3, damit Tabs als Leerzeichen an das Terminal gesendet werden, und HUPCL, damit die Leitung nach dem letzten Close getrennt wird.
<i>login-prompt</i>	Dieses gesamte Feld wird als Login-Prompt ausgegeben. Im Gegensatz zu den vorangehenden Feldern, in denen Leerzeichen, Tabs und Zeilenende-Zeichen ignoriert werden, werden sie in diesem Feld berücksichtigt.
<i>next-label</i>	Ist in diesem Feld nicht die gewünschte Übertragungsgeschwindigkeit angegeben, die dann zum Zuge kommt, wenn der Benutzer ein <break>-Zeichen eingibt, sucht getty nach einem Eintrag, dessen Label-Feld dem Inhalt von next-label entspricht und richtet das Terminal mit diesen Werten ein. Meist verknüpft man auf diese Weise eine Reihe von Übertragungsgeschwindigkeiten zu einer zyklisch geschlossenen Gruppe. Die Verknüpfung geht beispielsweise von 2400 zu 1200 zu 300 und von dort wieder zurück zu 2400.

Wird getty ohne zweites Argument aufgerufen, wird der erste Eintrag von /etc/gettydefs verwendet. Er wird auch benutzt, wenn getty das angegebene Label nicht finden kann. Ist /etc/gettydefs selbst nicht vorhanden, gibt es einen in das Kommando eingebauten Eintrag, der ein Terminal mit einer Geschwindigkeit von 300 Baud hochfährt.

Es wird dringend empfohlen, nach dem Erstellen oder Ändern von /etc/gettydefs das getty-Kommando mit der Option check auszuführen, um mögliche Fehler festzustellen.

Anhang 3: Dateiformate

DATEIEN:

/etc/gettydefs

SIEHE AUCH:

Administrator-Kommando: getty

Geräte-datei: termio

Benutzerkommando: login

System-schnittstelle: ioctl

Anhang 3: Dateiformate

group – Gruppendatei

BESCHREIBUNG:

Die Datei /etc/group enthält für jede Gruppe die folgenden Informationen:

- Gruppenname
- Verschlüsseltes Paßwort
- Numerische Gruppen-ID
- Liste aller der Gruppe angehörenden Benutzer, getrennt durch Kommata

Group ist eine ASCII-Datei. Die einzelnen Felder sind durch Doppelpunkte voneinander getrennt; jede Eintrag für eine Gruppe ist von der nächsten durch ein Zeilenende-Zeichen getrennt. Ist das Paßwortfeld leer, wird kein Paßwort verlangt.

Da die Paßwörter verschlüsselt sind, hat diese Datei allgemeine Leserlaubnis und dient z. B. dazu, numerischen Gruppen-IDs Namen zuzuordnen.

SIEHE AUCH:

- Dateiformat: passwd
- Benutzerkommandos: newgrp, passwd

Anhang 3: Dateiformate

inittab – Script für den init-Prozeß

BESCHREIBUNG:

Die inittab-Datei stellt das Script für die Rolle von init als allgemeinen Prozeßsteuerungsmechanismus bereit. Der Prozeß, der den größten Teil der Prozeßsteuerungsaktivitäten von init ausmacht, ist der Leitungsprozeß /etc/getty, der einzelne Terminalleitungen startet. Weitere Prozesse, die im typischen Fall von init gesteuert werden, sind Dämonen und die Shell.

Die Datei inittab besteht aus positionsabhängigen Einträgen mit dem folgenden Format:

Id:Rstatus:Aktion:Prozeß

Jeder Eintrag wird durch ein Zeilenende-Zeichen begrenzt. Steht vor dem Zeilenende-Zeichen jedoch ein Backslash (\), so wird der Eintrag in einer Folgezeile fortgesetzt. Ein Eintrag kann bis zu 512 Stellen lang sein. In das Prozeß-Feld können mittels der sh-Kommentarkonvention Kommentare eingefügt werden. Kommentare für Leitungen, die gettys erzeugen, werden vom Benutzerkommando who angezeigt. Die Kommentare sollten einige Informationen über die Leitung enthalten, z. B. den Ort. Die Datei inittab kann beliebig viele Einträge enthalten.

ID Hier stehen ein oder zwei Zeichen, die den Eintrag eindeutig kennzeichnen.

Rstatus Definiert den Run-Level, in dem dieser Eintrag verarbeitet werden soll. Run-Levels entsprechen effektiv einer Konfiguration von Prozessen im System. Das heißt, daß jedem von init erzeugten Prozeß ein oder mehrere Run-Levels zugeordnet werden, in denen er vorhanden sein darf. Die Run-Levels werden durch eine Zahl von 0 bis 6 dargestellt. Wenn sich das System beispielsweise im Run-Level 1 befindet, werden nur die Einträge verarbeitet, die eine 1 im Rstatus-Feld haben. Wird init aufgefordert, den Run-Level zu wechseln, wird an alle Prozesse, die im Rstatus-Feld keinen Eintrag für den neuen Run-Level haben, ein Warnsignal (SIGTERM) abgesetzt; nach 20 Sekunden werden diese Prozesse dann zwangsweise mit einem kill-Signal (SIGKILL) beendet. Im

Anhang 3: Dateiformate

Rstatus-Feld können für einen Prozeß mehrere Run-Levels definiert werden. Dazu werden mehrere Run-Levels von 0 bis 6 zusammen angegeben. Ist kein Run-Level angegeben, bedeutet dies, daß der Prozeß in allen Run-Levels von 0 bis 6 gültig ist. Es gibt drei weitere Werte, und zwar a, b und c, die im Rstatus-Feld stehen können, obwohl es sich nicht um echte Run-Levels handelt. Einträge mit diesen Zeichen im Rstatus-Feld werden nur dann verarbeitet, wenn der telinit-Prozeß (siehe Administrator-Kommando init) verlangt, daß sie ausgeführt werden (ungeachtet des aktuellen Run-Levels des Systems). Sie unterscheiden sich von Run-Levels dadurch, daß das System in diesem Zustand nur so lange verbleibt, bis alle zu den Zuständen gehörenden Einträge verarbeitet sind. Ein Prozeß, der von einem Kommando mit a, b oder c gestartet wurde, wird nicht mit kill zwangsweise beendet, wenn init den Run-Level wechselt. Sie werden nur dann gekillt, wenn ihre Leitung in /etc/inittab im Aktionsfeld als ausgeschaltet (off) markiert ist, wenn ihre Leitung vollständig aus /etc/inittab gelöscht ist oder wenn init in den SINGLE USER-Status übergeht.

Aktion

Schlüsselwörter in diesem Feld teilen init mit, wie der im Prozeß-Feld angegebene Prozeß zu behandeln ist. Init kennt folgende Aktionen:

- respawn Ist der entsprechende Prozeß nicht existent, den Prozeß starten, nicht auf seine Beendigung warten (weiterhin die inittab-Datei absuchen), und wenn er seine Existenz beendet, den Prozeß neu starten. Wenn der Prozeß bereits existiert, nichts unternehmen und die inittab-Datei weiterhin absuchen.
- wait Nachdem init in den Run-Level übergegangen ist, der dem Rstatus des Eintrags entspricht, den Prozeß starten und auf seine Beendigung warten. Jedesmal wenn inittab gelesen wird, solange init sich in demselben Run-Level befindet, ignoriert init diesen Eintrag.

Anhang 3: Dateiformate

once	Sobald init in einen Run-Level übergeht, der dem Rstatus des Eintrags entspricht, den Prozeß starten und nicht auf seine Beendigung warten. Wenn der Prozeß seine Existenz beendet, den Prozeß nicht neu starten. Nach dem Übergang in einen neuen Run-Level, in dem der Prozeß noch nach einem vorangehenden Run-Level-Wechsel läuft, wird das Programm nicht neu gestartet.
boot	Der Eintrag soll nur dann verarbeitet werden, wenn init die inittab-Datei beim Programmieren (Boot) liest. Init soll den Prozeß starten und nicht auf seine Beendigung warten; wenn der Prozeß seine Existenz beendet, soll es den Prozeß nicht neu starten. Damit diese Instruktion sinnvoll ist, muß der Rstatus den Standardwert haben oder er muß dem Run-Level von init zum Boot-Zeitpunkt entsprechen. Diese Aktion ist sinnvoll für eine Initialisierungsfunktion nach einem Reboot des Systems.
bootwait	Der Eintrag soll nur dann verarbeitet werden, wenn init die inittab-Datei beim Programmieren (Boot) liest. Init soll den Prozeß starten, auf seine Beendigung warten und, wenn der Prozeß seine Existenz beendet, diesen nicht neu starten.
powerfail	Der diesem Eintrag zugeordnete Prozeß soll nur ausgeführt werden, wenn init ein Netzausfallsignal (SIGPWR, siehe Systemschnittstelle signal) empfängt.
powerwait	Der diesem Eintrag zugeordnete Prozeß soll nur ausgeführt werden, wenn init ein Netzausfallsignal (SIGPWR) empfängt. Der Prozeß muß erst beendet sein, bevor die Verarbeitung von inittab fortgesetzt wird.
off	Wenn der diesem Eintrag zugeordnete Prozeß derzeit läuft, das Warnsignal (SIGTERM) aussenden und 20 Sekunden warten, bevor der Prozeß mit dem kill-Signal (SIGKILL) zwangsweise beendet wird. Wenn der Prozeß nicht existiert, den Eintrag ignorieren.

Anhang 3: Dateiformate

- ondemand** Diese Instruktion ist effektiv ein Synonym für die Aktion `respawn`. Sie hat dieselbe Funktion wie `respawn`, hat aber ein anderes Schlüsselwort, um die Assoziation mit den Run-Levels zu vermeiden. Diese Instruktion wird nur mit den Werten a, b und c im `Rstatus`-Feld verwendet.
- initdefault** Ein Eintrag mit dieser Aktion wird nur dann gelesen, wenn `init` aufgerufen wird. `init` stellt anhand dieses Eintrags, sofern er vorhanden ist, fest, in welchen Run-Level es anfangs übergehen soll. Dazu nimmt es den höchsten im `Rstatus`-Feld angegebenen Run-Level und verwendet diesen als anfänglichen Status. Wenn das `Rstatus`-Feld leer ist, wird dies als 0123456 interpretiert, so daß `init` in den Run-Level 6 übergeht. Im `initdefault`-Eintrag kann nicht angegeben werden, daß `init` im `SINGLE USER`-Status beginnen soll. Darüber hinaus gilt, daß `init` dann, wenn es keinen `initdefault`-Eintrag in `/etc/inittab` findet, vom Benutzer beim Reboot einen anfänglichen Run-Level verlangt.
- sysinit** Einträge dieses Typs werden ausgeführt, bevor `init` versucht, die Konsole anzusprechen. Es wird davon ausgegangen, daß dieser Eintrag nur dazu verwendet wird, Geräte zu initialisieren, auf denen `init` versuchen könnte, nach dem Run-Level zu fragen. Diese Einträge werden ausgeführt und ihre Beendigung abgewartet, bevor die Verarbeitung fortgesetzt wird.
- Prozeß** In diesem Feld steht ein auszuführendes `sh`-Kommando. Vor dem gesamten `Prozeß`-Feld steht `exec`, und es wird an eine mit `fork` erzeugte `sh` als `sh -c 'exec Kommando'` übergeben. Aus diesem Grund kann im `Prozeß`-Feld jede beliebige zulässige `sh`-Syntax erscheinen. Kommentare können Sie mit der Syntax `„; #Kommentar“` einfügen.

Anhang 3: Dateiformate

DATEIEN:

/etc/inittab

SIEHE AUCH:

Administrator-Kommandos: getty, init

Benutzerkommandos: sh, who

Systemschnittstellen: exec, open, signal

Anhang 3: Dateiformate

inode – Format eines I-Knotens

SYNTAX:

```
#include <sys/types.h>
#include <sys/ino.h>
```

BESCHREIBUNG:

Ein I-Knoten für eine gewöhnliche Datei oder ein Verzeichnis in einem Dateisystem hat die folgende von <sys/ino.h> definierte Struktur:

```
/* Inode structure as it appears on a disk block. */
struct dinode
{
    ushort di_mode;        /* mode and type of file */
    short  di_nlink;      /* number of links to file */
    ushort di_uid;        /* owner's user id */
    ushort di_gid;        /* owner's group id */
    off_t  di_size;       /* number of bytes in file */
    char   di_addr[40];   /* disk block addresses */
    time_t di_atime;      /* time last accessed */
    time_t di_mtime;      /* time last modified */
    time_t di_ctime;      /* time of last file status change */
};
/*
 * the 40 address bytes:
 * 39 used; 13 addresses
 * of 3 bytes each.
 */
```

DATEIEN:

/usr/include/sys/ino.h

SIEHE AUCH:

Systemschnittstelle: stat
Dateiformat: fs

Anhang 3: Dateiformate

issue – Datei für Projektbezeichnung

BESCHREIBUNG:

Die Datei /etc/issue enthält die Bezeichnung des Projekts oder Gegenstandes, der als Login-Prompt ausgegeben werden soll. Sie ist eine ASCII-Datei, die vom Programm getty gelesen und dann an das entsprechende Terminal ausgegeben wird.

DATEIEN:

/etc/issue

SIEHE AUCH:

Benutzerkommando: login

Anhang 3: Dateiformate

mnttab – Tabelle der eingehängten Dateisysteme

SYNTAX:

```
#include <mnttab.h>
```

BESCHREIBUNG:

Die Datei mnttab liegt im Verzeichnis /etc und enthält eine Tabelle der Geräte, die durch das mount-Kommando eingehängt wurden. Die Struktur dieser Tabelle ist folgendermaßen in <mnttab.h> definiert:

```
struct mnttab
{
    char    mt_dev[32];
    char    mt_filsys[32];
    short   mt_ro_flg;
    time_t  mt_time;
};
```

Jeder Eintrag hat eine Länge von 70 Bytes. Die ersten 32 Bytes enthalten die mit Nullen aufgefüllten Namen der Gerätedateien und die nächsten 32 Bytes enthalten die mit Nullen aufgefüllten Namen der entsprechenden Wurzelverzeichnisse. Die letzten sechs Bytes enthalten die Zugriffsrechte und das Datum, an dem das Dateisystem eingehängt wurde.

Die maximale Anzahl der Einträge in mnttab hängt von dem Systemparameter NMOUNT ab, der sich in /usr/src/uts/cf/conf.c befindet. Er enthält die Anzahl der maximal zulässigen Dateisysteme.

SIEHE AUCH:

Administrator-Kommandos: mount, setmnt

Anhang 3: Dateiformate

passwd – Paßwortdatei

BESCHREIBUNG:

Passwd enthält für jeden Benutzer die folgenden Informationen:

- Login-Name
- Verschlüsseltes Paßwort
- Numerische Benutzer-ID
- Numerische Gruppen-ID
- GCOS Jobnummer, Boxnummer, optionale GCOS Benutzer-ID
- Anfängliches aktuelles Verzeichnis
- Kommandointerpreter

Passwd ist eine ASCII-Datei. Die einzelnen Felder der Einträge sind durch einen Doppelpunkt voneinander getrennt. Das GCOS-Feld wird nur verwendet, wenn mit einem solchen System kommuniziert werden soll, und kann bei anderen Anlagen beliebigen Inhalt haben. Die Benutzereinträge sind voneinander durch ein Zeilenende-Zeichen getrennt. Ist das Paßwort-Feld leer, wird kein Paßwort verlangt; ist das Feld für den Kommandointerpreter leer, wird die Shell (/bin/sh) selbst verwendet.

Passwd liegt im Verzeichnis /etc. Da die Paßwörter verschlüsselt sind, hat sie allgemeine Leseerlaubnis.

Das verschlüsselte Paßwort besteht aus 13 Zeichen aus einem 64 Zeichen umfassenden Zeichensatz (., /, 0-9, A-Z, a-z). Eine zeitliche Begrenzung für die Paßwörter gilt für einen bestimmten Benutzer, wenn seinem verschlüsselten Paßwort in der Paßwortdatei ein Komma und eine nicht leere Zeichenkette mit Zeichen aus dem oben angegebenen Zeichensatz folgt. (Eine solche Zeichenkette muß beim ersten Mal vom Superuser editiert werden.)

Anhang 3: Dateiformate

Das erste Zeichen der zeitlichen Begrenzung, M , gibt die maximale Anzahl Wochen an, für die das Paßwort gültig ist. Versucht ein Benutzer sich anzumelden, nachdem sein Paßwort verfallen ist, so muß er ein neues Paßwort definieren. Das nächste Zeichen, m , gibt an, wie viele Wochen mindestens vergehen müssen, bevor das Paßwort geändert werden kann. Die übrigen Zeichen geben die Woche an (gezählt ab Anfang 1970), in der das Paßwort zuletzt geändert wurde. (Eine leere Zeichenkette ist gleichbedeutend mit Null.) M und m haben numerische Werte von 0–63, die dem oben genannten 64-Zeichen-Satz entsprechen (d. h. $1 = 1$ Woche; $z = 63$ Wochen). Ist $m = M = 0$ (abgeleitet von der Zeichenkette `.` oder `..`), wird der Benutzer gezwungen, sein Paßwort bei der nächsten Anmeldung zu ändern (und die zeitliche Begrenzung wird aus seinem Eintrag in der Paßwortdatei entfernt). Ist $m > M$ (z. B. dargestellt durch die Zeichenkette `./`), kann nur der Superuser das Paßwort ändern.

DATEIEN:

`/etc/passwd`

SIEHE AUCH:

Benutzerkommandos: `login`, `passwd`

Dateiformat: `group`

Anhang 3: Dateiformate

profile – Datei zum Einrichten der Benutzerumgebung

BESCHREIBUNG:

Wenn Ihr Login-Verzeichnis eine Datei mit dem Namen .profile enthält, wird diese Datei (über `exec .profile`) ausgeführt, bevor Ihre Terminalsituation beginnt; .profile-Dateien sind gut geeignet zum Setzen exportierter Umgebungsvariablen und von Terminalbetriebsarten. Wenn die Datei /etc/profile existiert, wird sie für jeden Benutzer vor der Datei .profile ausgeführt. Das folgende Beispiel ist typisch (abgesehen von den Kommentaren):

```
# Setzen globaler Umgebungsvariablen
export MAIL PATH TERM
# Setzen der Standardzugriffsrechte
umask 722
# Information über eingehende Post
MAIL=/usr/mail/doris
# Einbinden des eigenen /bin-Verzeichnisses in den Suchpfad der Shell
PATH=$PATH:$HOME/bin
```

DATEIEN:

```
$HOME/.profile
/etc/profile
```

SIEHE AUCH:

Benutzerkommandos: env, login, mail, sh, tty, su

Anhang 3: Dateiformate

security – Login-Sicherheitsmaßnahmen

BESCHREIBUNG:

Das Programm `/bin/login` stellt mehrere Sicherheitsmaßnahmen zur Verfügung:

- Paßwortänderung in gewissen Zeitabständen.
- Paßwortpflicht für alle Benutzer.
- Aufzeichnung von Fehlern bei der Anmeldung in `/usr/adm/badlog`.
- Anmeldeversuche eines Benutzers, die nicht erfolgreich verlaufen, werden auf die Anzahl von drei Versuchen beschränkt.
- Zur Autorisierung von Wählverbindungen wird ein gesondertes Paßwort verlangt.

Durch entsprechende Einträge in der Datei `/etc/security` kann der Administrator bestimmen, welche der o. g. Sicherheitsmaßnahmen durchgeführt werden. Diese Datei muß Root gehören und nur Root darf die Schreiberlaubnis besitzen. Standardmäßig wird die Datei so ausgeliefert, daß alle Sicherheitsvorkehrungen aktiv sind. Ist die Datei nicht vorhanden, sind ebenfalls alle Sicherheitsvorkehrungen aktiv. Ist die Datei zwar vorhanden, aber leer, beachtet das login-Programm keine dieser Sicherheitsmaßnahmen.

Folgende Einträge in `/etc/security` – die jeweils durch ein Zeilenende-Zeichen abgeschlossen sein müssen – werden von login erkannt:

AGEING	Paßwörter werden auf ihr Alter überprüft.
PASSREQ	Für jeden Benutzer ist ein Paßwort vorgeschrieben.
LOG_TO_FILE	Fehler beim Login werden in <code>/usradm/badlog</code> aufgezeichnet.
THREE_TRYS	Die Anzahl der nicht erfolgreichen Anmeldungen wird auf drei begrenzt.
DIALPASS	Für Wählverbindungen wird ein zusätzliches Paßwort verlangt.

Anhang 3: Dateiformate

Das Paßwort für Wählerverbindungen wird verschlüsselt und in der Datei /etc/d_passwd abgelegt. Ist diese Datei nicht vorhanden, findet keine Überprüfung des Paßworts für Wählerverbindungen statt.

DATEIEN:

/etc/security

SIEHE AUCH:

Benutzerkommando: login

Anhang 3: Dateiformate

term – Format der kompilierten Terminal-Datei

BESCHREIBUNG:

Kompilierte terminfo-Beschreibungen werden unter dem Verzeichnis `/usr/lib/terminfo` abgelegt. Um zu vermeiden, daß ein möglicherweise sehr umfangreiches Verzeichnis linear durchsucht werden muß, kommt ein zweistufiges Verfahren zur Anwendung: `/usr/lib/terminfo/c/Name`, wobei Name der Name des Terminals und c das erste Zeichen von Name ist. Somit wird `act4` in der Datei `/usr/lib/terminfo/a/act4` gefunden. Synonyme für das selbe Terminal werden durch mehrfache Verweise (Links) auf die selbe kompilierte Datei realisiert.

Das Format wurde so gewählt, daß es auf jeglicher Hardware identisch ist. Es wird von einem acht oder mehr Bit langen Byte ausgegangen, aber es werden keine Annahmen hinsichtlich der Byteanordnung oder Vorzeichenerweiterung gemacht.

Die kompilierte Datei wird mit dem `compile`-Programm erstellt und von der Routine `setupterm` gelesen. Diese beiden Softwarekomponenten sind Bestandteil von `curses`. Die Datei gliedert sich in sechs Teile: den Header, Terminalnamen, boolesche Merker, den numerischen Teil, den Zeichenkettenteil und die Zeichenkettentabelle.

Der Header-Teil steht am Anfang der Datei. Dieser Abschnitt enthält sechs short Integers in dem im folgenden beschriebenen Format. Bei diesen Ganzzahlen handelt es sich um:

1. die Magic Number (oktal 0432),
2. die Größe des Namensteils in Bytes,
3. die Anzahl der Bytes im booleschen Teil,
4. die Anzahl der short Integers im numerischen Teil,
5. die Anzahl der Offsets (short Integers) im Zeichenkettenteil und
6. die Größe der Zeichenkettentabelle in Bytes.

Anhang 3: Dateiformate

Short Integers werden in zwei Bytes je 8 Bit abgelegt. Das erste Byte enthält die niederwertigen 8 Bits des Wertes, und das zweite Byte enthält die höherwertigen 8 Bits. (Somit ist der dargestellte Wert $256 * \text{second} + \text{first}$.) Der Wert -1 wird durch `0377, 0377` dargestellt; andere negative Werte sind unzulässig. Die -1 bedeutet im allgemeinen, daß bei diesem Terminal eine Funktionsmöglichkeit nicht vorhanden ist. Dieses Format entspricht der Hardware der VAX und PDP-11. Maschinen, bei denen dieses Format nicht der Hardware entspricht, lesen die Ganzzahlen als zwei Bytes und errechnen das Ergebnis.

Als nächstes kommt der Abschnitt mit den Terminalnamen. Er enthält die erste Zeile der `terminfo`-Beschreibung, in der die verschiedenen Namen für das Terminal durch `|` getrennt aufgeführt sind. Dieser Abschnitt wird durch ein ASCII NUL-Zeichen abgeschlossen.

Die booleschen Merker sind in einem Byte pro Merker codiert. Das Byte ist entweder 0 oder 1 und bedeutet Merker vorhanden bzw. nicht vorhanden. Diese Funktionsmöglichkeiten sind in der selben Reihenfolge wie in der Datei `<term.h>` angegeben.

Zwischen dem booleschen Teil und dem numerischen Teil wird nötigenfalls ein Nullbyte eingefügt, damit der numerische Teil auf einem geraden Byte beginnt. Alle short Integers werden an einer Kurzwortgrenze ausgerichtet.

Der numerische Teil gleicht dem Merkerteil. Jede Funktionsmöglichkeit belegt zwei Bytes und wird als short Integer abgespeichert. Wenn der dargestellte Wert -1 ist, ist die entsprechende Funktionsmöglichkeit nicht vorhanden.

Der Zeichenkettenteil ist ebenfalls ähnlich aufgebaut. Jede Funktionsmöglichkeit ist als short Integer in dem oben beschriebenen Format abgelegt. Der Wert -1 bedeutet, daß die Funktionsmöglichkeit nicht vorhanden ist. Andernfalls wird der Wert als Offset ab Anfang der Zeichenkettentabelle verstanden. Sonderzeichen in der Schreibweise mit `~X` oder `\c` werden in ihrer interpretierten Form gespeichert und nicht in der druckbaren Darstellung. Auffüllinformationen (Zeitverzögerungen) `$<nn>` und Parameterinformationen `%x` werden in nicht interpretierter Form gespeichert.

Anhang 3: Dateiformate

Den letzten Abschnitt bildet die Zeichenkettentabelle. Sie enthält alle Werte von Zeichenketten-Funktionsmöglichkeiten, die im Zeichenkettenteil angesprochen werden. Jede Zeichenkette wird mit Nullzeichen abgeschlossen.

Es ist möglich, daß setupterm andere Funktionsmöglichkeiten erwartet als in der Datei effektiv vorhanden sind. Dies kann der Fall sein, wenn die Datenbasis seit der Rekompilierung von setupterm geändert wurde (was zu zusätzlichen, nicht erkannten Einträgen in der Datei führt) oder wenn das Programm später rekompiliert als die Datenbank geändert wurde (was zu fehlenden Einträgen führt). Die Routine setupterm muß auf beide Möglichkeiten vorbereitet werden, und deshalb sind auch die Zahlen und Größen darin vorhanden. Darüber hinaus müssen neue Funktionsmöglichkeiten immer an das Ende der Liste der booleschen, numerischen und Zeichenketten-Funktionsmöglichkeiten angefügt werden.

Als Beispiel folgt der oktale Speicherauszug der Beschreibung für das Terminal Microterm ACT 4:

```

microterm|act4|microterm act iv,
  cr=^M, cud1=^J, ind=^J, bel=^G, am, cub1=^H,
  ed=^, el=^, clear=^L, cup=^T%p1%c%p2%c,
  cols#80, lines#24, cuf1=^X, cuu1=^Z, home=^],

000 032 001      \0 025 \0 \b \0 212 \0 " \0 m i c r
020 o t e r m l a c t 4 l m i c r o
040 t e r m a c t i v \0 \0 001 \0 \0
060 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0
100 \0 \0 P \0 377 377 030 \0 377 377 377 377 377 377 377
120 377 377 377 377 \0 \0 002 \0 377 377 377 377 004 \0 006 \0
140 \b \0 377 377 377 377 \n \0 026 \0 030 \0 377 377 032 \0
160 377 377 377 377 034 \0 377 377 036 \0 377 377 377 377 377
200 377 377 377 377 377 377 377 377 377 377 377 377 377 377
*
520 377 377 377 377 \0 377 377 377 377 377 377 377 377 377
540 377 377 377 377 377 007 \0 \r \0 \f \0 036 \0 037 \0
560 024 % p 1 % c % p 2 % c \0 \n \0 035 \0
600 \b \0 030 \0 032 \0 \n \0

```

Anhang 3: Dateiformate

Die kompilierten Einträge dürfen insgesamt nicht länger als 4096 Bytes sein. Das Namenfeld kann maximal 128 Bytes lang sein.

DATEIEN:

`/usr/lib/terminfo/*/*` kompilierte Datenbasis der Terminalfunktionen

SIEHE AUCH:

Dateiformat: `terminfo`

Anhang 3: Dateiformate

terminfo – Datenbasis der Terminalfunktionen

BESCHREIBUNG:

Terminfo ist eine Datenbasis, in der Terminals beschrieben sind. Diese Daten werden beispielsweise von vi und curses benutzt. Die Terminals werden in terminfo durch Angabe der vorhandenen Funktionsmöglichkeiten und der Art und Weise, wie Operationen ausgeführt werden, beschrieben. Ferner sind in terminfo die Bedingungen für Zeitverzögerungen (padding) und die Initialisierungsfolgen angegeben.

Die Einträge in terminfo bestehen aus mehreren Feldern, die durch Kommata voneinander getrennt sind. Leerstellen nach einem Komma werden ignoriert. Der erste Eintrag für ein Terminal enthält die für das Terminal bekannten Namen, die durch '|' voneinander getrennt sind. Der erste angegebene Name ist die gebräuchlichste Abkürzung für das Terminal, der letzte Name ist eine ausführliche, genaue Bezeichnung des Terminals, und alle anderen Namen sind Synonyme für den Terminalnamen. Alle Namen mit Ausnahme des letzten müssen aus Kleinbuchstaben bestehen und dürfen keine Leerzeichen enthalten; der letzte Name kann zur besseren Lesbarkeit Großbuchstaben und Leerzeichen enthalten.

Die Terminalnamen (mit Ausnahme des letzten ausführlichen Eintrags) sollten nach folgenden Konventionen festgelegt werden. Für die Hardware, aus der das Terminal besteht, ist ein Wurzelname zu wählen, z. B. dab4x. Der Name soll keine Bindestriche enthalten außer in Synonymen, die so gewählt werden können, daß keine Überschneidungen mit anderen Namen vorkommen. Betriebsarten der Hardware oder benutzerspezifische Merkmale sind anzugeben, indem ein Bindestrich und eine Kennzeichnung der Betriebsart an den Namen angehängt wird. Ein vt100-Terminal im 132-Spalten-Modus würde man etwa als vt100-w bezeichnen. Wann immer möglich sollten die folgenden Suffixe verwendet werden:

Anhang 3: Dateiformate

Suffix	Bedeutung	Beispiel
-w	Breiter Modus (wide) (mehr als 80 Spalten)	vt100-w
-am	Mit automatischer Zeilenschaltung (normalerweise Standard)	vt100-am
-nam	Ohne automatische Zeilenschaltung	vt100-nam
-n	Anzahl Zeilen auf dem Bildschirm	aaa-60
-na	Keine Pfeiltasten (im lokalen Modus lassen)	c100-na
-np	Anzahl der Speicherseiten	c100-4p
-rv	Invertierte Darstellung (reverse video)	c100-rv

FUNKTIONSMÖGLICHKEITEN:

Die Variable ist der Name, unter dem der Programmierer (auf der term-
info-Ebene) die Funktionsmöglichkeit anspricht. Der capname (Name
für die Funktionsmöglichkeit) ist die im Text der Datenbasis verwen-
dete Kurzbezeichnung. Dieser Name wird von demjenigen verwendet, der
die Datenbasis aktualisiert. Der i.code ist der aus zwei Buchstaben be-
stehende interne Code, der in der kompilierten Datenbasis verwendet
wird. Er entspricht immer dem alten termcap-Funktionsnamen.

Für die Funktionsnamen gibt es keine feste Längenbegrenzung, aber
im allgemeinen hält man sich an eine maximale Länge von fünf Stellen,
damit die Namen nicht ausufern und die Tabs in der Quelldatei caps
gleichmäßig und übersichtlich angeordnet sind. Die Namen werden
wann immer möglich so gewählt, daß sie dem ANSI-Standard X3.64-
1979 entsprechen oder ähnlich sind. Die Semantik soll auch derjenigen
der Spezifikation entsprechen.

- (P) bedeutet, daß eine Zeitverzögerung (padding) spezifiziert werden kann.
- (G) bedeutet, daß die Zeichenkette durch tparm mit den angegebenen Parametern verarbeitet wird (#_i).
- (*) bedeutet, daß ein Auffüllen entsprechend der Anzahl der betroffenen Zeilen erfolgen kann.
- (#_i) gibt den i-ten Parameter an.

Anhang 3: Dateiformate

Variable	Cap-Name	I. Code	Beschreibung
Booleans			
auto_left_margin,	bw	bw	cut1 wraps from column 0 to last column
auto_right_margin,	am	am	Terminal has automatic margins
beehive_glitch,	xsb	xb	Beehive (f1=escape, f2=ctrl C)
ceol_standout_glitch,	xhp	xs	Standout not erased by overwriting (hp)
eat_newline_glitch,	xenl	xn	newline ignored after 80 cols (Concept)
erase_overstrike,	eo	eo	Can erase overstrikes with a blank
generic_type,	gn	gn	Generic line type (e.g., dialup, switch).
hard_copy,	hc	hc	Hardcopy terminal
has_meta_key,	km	km	Has a meta key (shift, sets parity bit)
has_status_line,	hs	hs	Has extra "status line"
insert_null_glitch,	in	in	Insert mode distinguishes nulls
memory_above,	da	da	Display may be retained above the screen
memory_below,	db	db	Display may be retained below the screen
move_insert_mode,	mir	mi	Safe to move while in insert mode
move_standout_mode,	msgr	ms	Safe to move in standout modes
over_strike,	os	os	Terminal overstrikes
status_line_esc_ok,	eslok	es	Escape can be used on the status line
teleray_glitch,	xt	xt	Tabs ruin, magic so char (Teleray 1061)
tilde_glitch,	hz	hz	Hazeltine; can not print ~'s
transparent_underline,	ul	ul	underline character overstrikes
xon_xoff,	xon	xo	Terminal uses xon/xoff handshaking
Zahlen:			
columns,	cols	co	Number of columns in a line
init_tabs,	it	it	Tabs initially every # spaces
lines,	lines	li	Number of lines on screen or page
lines_of_memory,	lm	lm	Lines of memory if > lines. 0 means varies
magic_cookie_glitch,	xmc	sg	Number of blank chars left by smso or rmso
padding_baud_rate,	pb	pb	Lowest baud where cr/nl padding is needed
virtual_terminal,	vt	vt	Virtual terminal number (UNIX system)
width_status_line,	wsl	ws	No. columns in status line
Zeichenketten:			
back_tab,	cbt	bt	Back tab (P)
bell,	bel	bl	Audible signal (bell) (P)
carriage_return,	cr	cr	Carriage return (P*)
change_scroll_region,	csr	cs	change to lines #1 through #2 (vt100) (PG)
clear_all_tabs,	tbc	ct	Clear all tab stops (P)
clear_screen,	clear	cl	Clear screen and home cursor (P*)
clr_eol,	el	ce	Clear to end of line (P)
clr_eos,	ed	cd	Clear to end of display (P*)
column_address,	hpa	ch	Set cursor column (PG)
command_character,	cmdch	CC	Term. settable cmd char in prototype
cursor_address,	cup	cm	Screen rel. cursor motion row #1 col #2 (PG)

Anhang 3: Dateiformate

cursor_down,	cursor_home,	cursor_invisible,	cursor_left,	cursor_mem_address,	cursor_normal,	cursor_right,	cursor_to_ll,	cursor_up,	cursor_visible,	delete_character,	delete_line,	dis_status_line,	down_half_line,	enter_alt_charset_mode,	enter_blink_mode,	enter_bold_mode,	enter_ca_mode,	enter_delete_mode,	enter_dim_mode,	enter_insert_mode,	enter_protected_mode,	enter_reverse_mode,	enter_secure_mode,	enter_standout_mode,	enter_underline_mode,	erase_chars	exit_alt_charset_mode,	exit_attribute_mode,	exit_ca_mode,	exit_delete_mode,	exit_insert_mode,	exit_standout_mode,	exit_underline_mode,	flash_screen,	form_feed,	from_status_line,	init_1string,	init_2string,	init_3string,	init_file,	insert_character,	insert_line,	insert_padding,	key_backspace,	key_catab,	key_clear,	cud1 do home ho civis vi cub1 le mrcup CM cnorm ve cuf1 nd ll ll cuu1 up cvvis vs dch1 dc dl1 dl dsl ds hd hd smacs as blink mb bold md smcup ti smdc dm dim mh smir im prot mp rev mr invis mk smso so smul us ech ec rmacs ae sgr0 me mrcup te rmdc ed rmir ei rmso se rmul ue flash vb ff ff fsl fs is1 i1 is2 i2 is3 i3 if if ich1 ic il1 al ip ip kbs kb ktbc ka kclr kc	Down one line Home cursor (if no cup) Make cursor invisible Move cursor left one space Memory relative cursor addressing Make cursor appear normal (undo vs/vi) Non-destructive space (cursor right) Last line, first column (if no cup) Upline (cursor up) Make cursor very visible Delete character (P*) Delete line (P*) Disable status line Half-line down (forward 1/2 linefeed) Start alternate character set (P) Turn on blinking Turn on bold (extra bright) mode String to begin programs that use cup Delete mode (enter) Turn on half-bright mode Insert mode (enter); Turn on protected mode Turn on reverse video mode Turn on blank mode (chars invisible) Begin stand out mode Start underscore mode Erase #1 characters (PG) End alternate character set (P) Turn off all attributes String to end programs that use cup End delete mode End insert mode End stand out mode End underscore mode Visible bell (may not move cursor) Hardcopy terminal page eject (P*) Return from status line Terminal initialization string Terminal initialization string Terminal initialization string Name of file containing is Insert character (P) Add new blank line (P*) Insert pad after character inserted (p*) Sent by backspace key Sent by clear-all-tabs key Sent by clear screen or erase key
--------------	--------------	-------------------	--------------	---------------------	----------------	---------------	---------------	------------	-----------------	-------------------	--------------	------------------	-----------------	-------------------------	-------------------	------------------	----------------	--------------------	-----------------	--------------------	-----------------------	---------------------	--------------------	----------------------	-----------------------	-------------	------------------------	----------------------	---------------	-------------------	-------------------	---------------------	----------------------	---------------	------------	-------------------	---------------	---------------	---------------	------------	-------------------	--------------	-----------------	----------------	------------	------------	---	---

© „Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Mitteilung ihres Inhalts nicht gestattet, soweit nicht ausdrücklich zugelassen. Insbesondere ist die Weitergabe oder Vervielfältigung dieser Unterlage ohne die vorherige schriftliche Genehmigung der Nixdorf Computer AG nicht zulässig.“

A3

Anhang 3: Dateiformate

key_ctab,	kctab	kt	Sent by clear-tab key
key_dc,	kdch1	kD	Sent by delete character key
key_dl,	kd1	kL	Sent by delete line key
key_down,	kcud1	kd	Sent by terminal down arrow key
key_elc,	krmir	kM	Sent by rmir or smir in insert mode
key_eol,	kel	kE	Sent by clear-to-end-of-line key
key_eos,	ked	kS	Sent by clear-to-end-of-screen key
key_f0,	kf0	k0	Sent by function key f0
key_f1,	kf1	k1	Sent by function key f1
key_f10,	kf10	ka	Sent by function key f10
key_f2,	kf2	k2	Sent by function key f2
key_f3,	kf3	k3	Sent by function key f3
key_f4,	kf4	k4	Sent by function key f4
key_f5,	kf5	k5	Sent by function key f5
key_f6,	kf6	k6	Sent by function key f6
key_f7,	kf7	k7	Sent by function key f7
key_f8,	kf8	k8	Sent by function key f8
key_f9,	kf9	k9	Sent by function key f9
key_home,	khome	kh	Sent by home key
key_ic,	kich1	kl	Sent by ins char/enter ins mode key
key_il,	ki1	kA	Sent by insert line
key_left,	kcub1	kl	Sent by terminal left arrow key
key_ll,	ki1	kH	Sent by home-down key
key_npage,	knp	kN	Sent by next-page key
key_ppage,	kpp	kP	Sent by previous-page key
key_right,	kcu1	kr	Sent by terminal right arrow key
key_sf,	kind	kF	Sent by scroll-forward/down key
key_sr,	kri	kR	Sent by scroll-backward/up key
key_stab,	khts	kT	Sent by set-tab key
key_up,	kcu1	ku	Sent by terminal up arrow key
key_padlocal,	rmkx	ke	Out of "keypad transmit" mode
key_padxmit,	smkx	ks	Put terminal in "keypad transmit" mode
lab_f0,	lf0	l0	Labels on function key f0 if not f0
lab_f1,	lf1	l1	Labels on function key f1 if not f1
lab_f10,	lf10	la	Labels on function key f10 if not f10
lab_f2,	lf2	l2	Labels on function key f2 if not f2
lab_f3,	lf3	l3	Labels on function key f3 if not f3
lab_f4,	lf4	l4	Labels on function key f4 if not f4
lab_f5,	lf5	l5	Labels on function key f5 if not f5
lab_f6,	lf6	l6	Labels on function key f6 if not f6
lab_f7,	lf7	l7	Labels on function key f7 if not f7
lab_f8,	lf8	l8	Labels on function key f8 if not f8
lab_f9,	lf9	l9	Labels on function key f9 if not f9
meta_on,	smm	mm	Turn on "meta mode" (8th bit)
meta_off,	rmr	mo	Turn off "meta mode"
newline,	nel	nw	Newline (behaves like cr followed by lf)
pad_char,	pad	pc	Pad character (rather than null)

Anhang 3: Dateiformate

parm_dch,	dch	DC	Delete #1 chars (PG*)
parm_delete_line,	dl	DL	Delete #1 lines (PG*)
parm_down_cursor,	cud	DO	Move cursor down #1 lines (PG*)
parm_ich,	ich	IC	Insert #1 blank chars (PG*)
parm_index,	indn	SF	Scroll forward #1 lines (PG)
parm_insert_line,	il	AL	Add #1 new blank lines (PG*)
parm_left_cursor,	cub	LE	Move cursor left #1 spaces (PG)
parm_right_cursor,	cuf	RI	Move cursor right #1 spaces (PG*)
parm_rindex,	rin	SR	Scroll backward #1 lines (PG)
parm_up_cursor,	cuu	UP	Move cursor up #1 lines (PG*)
pkey_key,	pfkey	pk	Prog funct key #1 to type string #2
pkey_local,	pfloc	pl	Prog funct key #1 to execute string #2
pkey_xmit,	pfx	px	Prog funct key #1 to xmit string #2
print_screen,	mc0	ps	Print contents of the screen
prtr_off,	mc4	pf	Turn off the printer
prtr_on,	mc5	po	Turn on the printer
repeat_char,	rep	rp	Repeat char #1 #2 times. (PG*)
reset_1string,	rs1	r1	Reset terminal completely to sane modes.
reset_2string,	rs2	r2	Reset terminal completely to sane modes.
reset_3string,	rs3	r3	Reset terminal completely to sane modes.
reset_file,	rf	rf	Name of file containing reset string
restore_cursor,	rc	rc	Restore cursor to position of last sc
row_address,	vpa	cv	Vertical position absolute (set row) (PG)
save_cursor,	sc	sc	Save cursor position (P)
scroll_forward,	ind	sf	Scroll text up (P)
scroll_reverse,	ri	sr	Scroll text down (P)
set_attributes,	sgr	sa	Define the video attributes (PG9)
set_tab,	hts	st	Set a tab in all rows, current column
set_window,	wind	wi	Current window is lines #1-#2 cols #3-#4
tab,	ht	ta	Tab to next 8 space hardware tab stop
to_status_line,	tsl	ts	Go to status line, column #1
underline_char,	uc	uc	Underscore one char and move past it
up_half_line,	hu	hu	Half-line up (reverse 1/2 linefeed)
init_prog,	iprog	iP	Path name of program for init
key_a1,	ka1	K1	Upper left of keypad
key_a3,	ka3	K3	Upper right of keypad
key_b2,	kb2	K2	Center of keypad
key_c1,	kc1	K4	Lower left of keypad
key_c3,	kc3	K5	Lower right of keypad
prtr_non,	mc5p	pO	Turn on the printer for #1 bytes

Anhang 3: Dateiformate

Beispiel eines Eintrags

Der folgende Eintrag, der die Concept-100 beschreibt, gehört derzeit zu den komplizierteren Einträgen in der terminfo-Datei.

```
concept100|c100|concept|c104|c100-4p|concept 100,
  am, bel=^G, blank=^EH, blink=^EC, clear=^L$<2*>, cnorm=^EW,
  cols#80, cr=^M$<9>, cub1=^H, cud1=^J, cuf1=^E=,
  cup=^Ea%p1% ' %+%c%p2% ' %+%c,
  cuu1=^E; , cvvis=^EW, db, dch1=^EA$<16*>, dim=^EE, dl1=^EB$<3*>,
  ed=^E^C$<16*>, el=^E^U$<16>, eo, flash=^Ek$<20>^EK, ht=^t$<8>,
  il1=^E^R$<3*>, in, ind=^J, .ind=^J$<9>, ip=$<16*>,
  is2=^EU^Ef^E7^E5^E8^E1^ENH^EK^E^200^Eo&^200^Eo^47^E,
  kbs=^h, kcub1=^E>, kcuu1=^E<, kcuu1=^E=, kcuu1=^E; ,
  kf1=^E5, kf2=^E6, kf3=^E7, khome=^E?,
  lines#24, mir, pb#9600, prot=^EI, rep=^Er%p1%&p2% ' %+%c$< 2*>,
  rev=^ED, rmcup=^Ev $<6>^Ep^r^n, rmir=^E^200, rmkx=^EX,
  rmso=^Ed^Ee, rmul=^Eg, rmul=^Eg, sgr0=^EN^200,
  smcup=^EU^Ev 8p^Ep^r, smir=^E^P, smkx=^EX, smso=^EE^ED,
  smul=^EG, tabs, ul, vt#8, xenl,
```

Die Einträge können in mehreren Folgezeilen fortgesetzt werden. Dazu muß man am Anfang jeder Zeile mit Ausnahme der ersten Leerzeichen eingeben. In den Zeilen können Kommentare stehen, die mit # eingeleitet werden müssen. In terminfo gibt es drei Arten von Funktionsmöglichkeiten: boolesche Funktionsmöglichkeiten, die angeben, daß das Terminal ein bestimmtes Merkmal besitzt, numerische Funktionsmöglichkeiten, die die Größe des Terminals oder bestimmte Verzögerungen angeben und Zeichenketten-Funktionsmöglichkeiten, in denen eine Zeichenfolge angegeben ist, mit der bestimmte Terminaloperationen ausgeführt werden.

Arten von Funktionsmöglichkeiten

Alle Funktionsmöglichkeiten haben Namen. Beispielsweise wird die Tatsache, daß die Concept über automatische Zeilenschaltung verfügt (d. h. bei Erreichen des Zeilenendes erfolgt ein automatischer Wagenrücklauf mit Zeilenvorschub), durch die Funktionsmöglichkeit am bezeichnet. Daher ist in der Beschreibung der Concept die Funktions-

Anhang 3: Dateiformate

möglichkeit am angegeben. Bei numerischen Funktionsmöglichkeiten wird hinter dem Namen das Zeichen # und dann der Wert angegeben. So hat beispielsweise cols, das die Anzahl der Zeichenpositionen auf dem Terminal angibt, bei der Concept den Wert 80.

Schließlich werden Funktionsmöglichkeiten als Zeichenketten, z. B. el (Folge für Löschen bis Ende Zeile) durch den zweistelligen Code, ein = und eine Zeichenkette, die bei dem nächsten Komma endet, angegeben. Bei einer solchen Funktionsmöglichkeit kann an beliebiger Stelle eine Zeitverzögerung in Millisekunden in der Form \$<...> angegeben werden, z. B. el=\EK\$<3>, und Füllzeichen werden von tputs bereitgestellt, damit diese Verzögerung eintritt. Die Verzögerung kann in Form einer Zahl, z. B. 20, oder in Form einer Zahl mit einem nachfolgenden *, d. h. 3*, angegeben werden. Ein * bedeutet, daß ein Auffüllen mit Füllzeichen proportional zur Anzahl der von der Operation betroffenen Zeilen erforderlich ist, und der angegebene Betrag gibt die Auffüllmenge pro betroffener Einheit an. (Beim Einfügen von Zeichen bezieht sich der Faktor trotzdem auf die Anzahl der betroffenen Zeilen. Diese beträgt immer 1, es sei denn, das Terminal besitzt xenl und die Software benutzt diese Funktion.) Wird ein * angegeben, ist es manchmal sinnvoll, eine Verzögerung in der Form 3.5 einzugeben, wenn man eine Verzögerung auf zehntel Millisekunden genau angeben möchte. (Es ist nur eine Nachkommastelle erlaubt.)

Bei den Funktionsmöglichkeiten mit Zeichenkette als Wert gibt es eine Reihe von Escape-Folgen, die das Codieren von Zeichen erleichtern. Die Zeichenfolgen \E und \e bedeuten ein Escape-Zeichen ^x bedeutet ein Control-x für ein beliebiges geeignetes x, und die Folgen \n \l \r \t \b \f \s bedeuten Zeilenende, Zeilenvorschub, Wagenrücklauf, Tab, Rückschritt, Formularvorschub bzw. Leerzeichen. Weitere Escape-Folgen sind \^ für ^, \\ für \, \, für Komma, \: für : und \0 für Null. (\0 erzeugt \200, das eine Zeichenkette nicht beendet, sondern auf den meisten Terminals ein Nullzeichen darstellt. Schließlich können Zeichen nach einem \ auch als drei Oktalziffern angegeben werden.

Manchmal müssen einzelne Funktionsmöglichkeiten als Kommentar ausgeklammert werden. Dazu muß ein Punkt vor den Namen der Funktionsmöglichkeit gesetzt werden. Siehe dazu das zweite ind in dem Beispiel oben.

© Weitergabe sowie Verwertung dieser Unterlagen, Verwertung und Weitergabe dieser Unterlagen, ist ohne schriftliche Genehmigung des Nixdorf Computer AG. Zuwiderhandlungen verpflichten zu Schadenersatz. Alle Rechte für den Fall der Patenterteilung oder Gebrauchsmustereintragung vorbehalten.

Anhang 3: Dateiformate

Erstellung von Beschreibungen

Im folgenden wird erläutert, wie Sie Terminalbeschreibungen erstellen. Eine Terminalbeschreibung läßt sich am rationellsten erstellen, indem Sie die Beschreibung eines ähnlichen Terminals in terminfo imitieren und die Beschreibung nach und nach aufbauen, wobei Sie Ausschnitte der Beschreibungen mit vi auf Korrektheit prüfen. Sie sollten auch bedenken, daß die terminfo-Datei möglicherweise nicht ausreicht, ein sehr ungewöhnliches Terminal zu beschreiben, und man kann unter Umständen sogar auf Fehler im vi-Editor stoßen. Um eine neue Terminalbeschreibung auf einfache Weise zu testen, können Sie der Umgebungsvariablen TERMINFO den Pfadnamen eines Verzeichnisses geben, das die kompilierte Beschreibung, an der Sie arbeiten, enthält. Programme greifen dann hierauf zurück statt auf /usr/lib/terminfo. Um den richtigen Wert für die Zeitverzögerung (padding) beim Einfügen von Zeilen herauszufinden (wenn der Terminalhersteller den Wert nicht dokumentiert hat), kann man folgenden Test durchführen: Editieren Sie eine Datei mit 9600 Baud, löschen Sie ungefähr 16 Zeilen mitten aus dem Bildschirm und betätigen Sie irgendeine Taste mehrere Male schnell hintereinander. Wenn dadurch der Bildschirm durcheinander gerät, muß normalerweise mehr zeitverzögert werden. Einen ähnlichen Test kann man für das Einfügen von Zeichen durchführen.

Grundlegende Funktionsmöglichkeiten

Die Anzahl der Zeichenpositionen (Spalten) pro Zeile wird von der numerischen Funktionsmöglichkeit cols angegeben. Wenn es sich bei dem Terminal um einen Bildschirm handelt, gibt die Funktionsmöglichkeit lines die Anzahl der Zeilen auf dem Bildschirm an. Springt der Cursor bei Erreichen des rechten Randes automatisch zum Anfang der nächsten Zeile, hat es die Funktionsmöglichkeit am. Wenn das Terminal seinen Bildschirm löschen kann, wobei der Cursor in die rechte obere Ecke (Home-Position) springt, wird dies durch die Zeichenketten-Funktionsmöglichkeit clear angegeben. Werden Zeichen nur überschrieben (die alten Zeichen werden nicht gelöscht), hat es die Funktionsmöglichkeit os. Handelt es sich bei dem Terminal um ein Druckerterminal ohne Softcopy-Einheit, geben Sie hc und os an. (os bezieht sich auf Speicherbildschirmterminals, z. B. TEKTRONIX 4010 und auf Hardcopy- und APL-Terminals.)

Anhang 3: Dateiformate

Gibt es einen Code, mit dem der Cursor um eine Position nach links bewegt wird (wie bei Rückschritt), geben Sie diese Funktionsmöglichkeit mit `cub1` an. Analog werden Codes, mit denen der Cursor nach rechts, nach oben und nach unten versetzt wird, als `cuf1`, `cuu1` bzw. `cod1` angegeben. Diese lokalen Cursorbewegungen dürfen den Text, den sie berühren, nicht verändern. Normalerweise wird man beispielsweise nicht `'cuf1= '` angeben, weil das Leerzeichen das Zeichen, über das der Cursor bewegt wurde, löschen würde.

In diesem Zusammenhang ist es sehr wichtig zu wissen, daß die in `terminfo` codierten lokalen Cursorbewegungen am linken und oberen Rand eines Bildschirmterminals nicht definiert sind. Ein Programm darf nie versuchen, den Cursor über den linken Rand hinaus nach links zu versetzen, außer wenn `bw` angegeben ist, und es darf nie versuchen, lokal über den oberen Rand hinauszugehen. Um Text aufwärts zu scrollen, muß das Programm den Cursor in die untere linke Ecke des Bildschirms setzen und die Zeichenkette `ind` (`index`) absetzen.

Um Text abwärts zu scrollen, setzt das Programm den Cursor in die obere linke Ecke des Bildschirms und setzt die Zeichenkette `ri` (`reverse index`) ab. Die Zeichenketten `ind` und `ri` sind nicht definiert, wenn sie nicht in den richtigen Ecken des Bildschirms abgesetzt werden.

Parametergesteuerte Versionen der Kommandos für Scrolling sind `indn` und `rin`, die die selbe Semantik wie `ind` und `ri` haben. Sie besitzen allerdings einen Parameter und scrollen den Bildschirm um die entsprechende Anzahl Zeilen. Auch sie sind nur in der entsprechenden Ecke des Bildschirms definiert.

Die Funktionsmöglichkeit `am` gibt an, ob der Cursor beim Ausgeben von Text am rechten Rand des Bildschirms stehenbleibt, aber dies gilt nicht notwendigerweise für ein `cuf1` von der letzten Zeichenposition aus. Es gibt nur eine einzige definierte lokale Bewegung vom linken Rand aus, und zwar dann, wenn `bw` angegeben ist; in diesem Fall bewirkt ein `cub1` vom linken Rand aus, daß der Cursor auf den rechten Rand der vorangehenden Zeile springt. Wenn `bw` nicht angegeben ist, ist der Effekt nicht definiert. Dies ist beispielsweise sinnvoll zum Zeichnen eines Rahmens um den Bildschirmrand herum. Wenn die automatische Zeilenschaltung am Terminal per Schalter ein- und ausgeschaltet werden kann, wird in der `terminfo`-Datei davon ausgegangen, daß diese

Anhang 3: Dateiformate

Funktion eingeschaltet ist, d. h. am. Wenn das Terminal über ein Kommando verfügt, das den Cursor auf die erste Stelle der nächsten Zeile versetzt, kann dieses Kommando mit `nel` (newline) angegeben werden. Es spielt keine Rolle, ob das Kommando den Rest der aktuellen Zeile löscht; wenn das Terminal kein `cr` und `lf` besitzt, ist es unter Umständen trotzdem möglich, aus einer dieser Funktionsmöglichkeiten oder beiden ein funktionierendes `nel` zu konstruieren.

Parametrisierte Zeichenketten

Die Cursoradressierung und andere Zeichenketten, die im Terminal mit Parametern ausgestattet werden müssen, werden durch eine parametrisierte Zeichenketten-Funktionsmöglichkeit beschrieben, die `printf`-ähnliche Escapes `%x` enthält. Für die Cursoradressierung wird die Funktionsmöglichkeit `cup` mit den beiden Parametern Zeile und Spalte angegeben. (Zeilen und Spalten werden von Null aufwärts numeriert und beziehen sich auf den für den Benutzer physisch sichtbaren Bildschirm, nicht auf einen unsichtbaren Bildschirmspeicher.) Wenn das Terminal über eine speicherbezogene Cursoradressierung verfügt, kann dies mit `mrucup` angegeben werden.

Der Parametermechanismus benutzt einen Speicher und manipuliert ihn mit speziellen `%`-Codes. Im typischen Fall legt eine Sequenz einen der Parameter in dem Stack ab und gibt ihn in einem festgelegten Format aus. Häufig sind kompliziertere Operationen notwendig.

Die `%`-Codes haben die folgende Bedeutung:

<code>%%</code>	gibt <code>%</code> aus
<code>%d</code>	<code>pop()</code> ausgeben wie bei <code>printf</code>
<code>%2d</code>	<code>pop()</code> ausgeben wie bei <code>%2d</code>
<code>%3d</code>	<code>pop()</code> ausgeben wie bei <code>%3d</code>
<code>%02d</code>	
<code>%03d</code>	wie bei <code>printf</code>
<code>%c</code>	<code>pop()</code> ausgeben ergibt <code>%c</code>

Anhang 3: Dateiformate

<code>%s</code>	<code>pop()</code> ausgeben ergibt <code>%s</code>
<code>%p[1-9]</code>	i-ten Parameter im Speicher ablegen
<code>%P[a-z]</code>	Variable <code>[a-z]</code> auf <code>pop()</code> setzen
<code>%g[a-z]</code>	Variable <code>[a-z]</code> holen und im Speicher ablegen
<code>%'c'</code>	Zeichenkonstante <code>c</code>
<code>%{nn}</code>	Ganzzahl-Konstante <code>nn</code>
<code>%</code>	<code>%- %* %/ %m</code>
<code>%& % %~</code>	Bit-Operationen: <code>push(pop() op pop())</code>
<code>%= %> %<</code>	logische Operationen: <code>push(pop() op pop())</code>
<code>%! %~</code>	unäre Operationen <code>push(pop() op pop())</code>
<code>%i</code>	1 zu den ersten beiden Parametern addieren (bei ANSI-Terminals)
<code>%? expr %t thenpart %e elsepart%;</code>	if-then-else, <code>%e</code> elsepart else-if's sind möglich wie bei Algol 68: <code>%? c₁ %t b₁ %e c₂ %t b₂ %e c₃ %t b₃ %e c₄ %t b₄ %e %;</code> <code>c_i</code> sind Bedingungen, <code>b_i</code> sind Rümpfe.

Binäre Operationen haben die postfix-Form, und die Operanden haben die übliche Reihenfolge. Um `x-5` zu erhalten, würde man also `„%gx%55)%-“` angeben.

Verfügt das Terminal über absolute Cursoradressierung für Zeilen und Spalten, können diese Werte als Einparameter-Funktionsmöglichkeiten `hpa` (horizontale Position absolut) und `vpa` (vertikale Position absolut) angegeben werden. Manchmal sind diese kürzer als die allgemeineren Folgen mit zwei Parametern (wie beim `hp2645`) und können vorzugsweise anstelle von `cup` verwendet werden. Wenn es parametergesteuerte lokale Cursorbewegungen (z. B. Bewegung um `n` Stellen nach rechts) gibt, können diese durch `cud`, `cub`, `cuf` und `cuu` mit einem einzelnen Parameter, der die Anzahl der Stellen angibt, angegeben werden. Dies ist hauptsächlich dann sinnvoll, wenn das Terminal `cup` besitzt, wie etwa die `TEKTRONIX 4025`.

Anhang 3: Dateiformate

Cursorbewegungen

Wenn das Terminal in der Lage ist, den Cursor direkt in die Ausgangsposition (die obere linke Ecke des Bildschirms) zu bringen, kann dies durch home angegeben werden; die Möglichkeit, den Cursor direkt in die untere linke Ecke zu bringen, wird mit ll angegeben; in diesem Zusammenhang wird der Cursor möglicherweise mit cuu von der Ausgangsposition nach oben bewegt werden, aber ein Programm sollte dies nie von sich aus tun (außer wenn ll dies tut), da das Programm nicht wissen kann, welche Folgen eine Aufwärtsbewegung des Cursors aus der Ausgangsposition hat. Beachten Sie, daß die Ausgangsposition nicht dasselbe ist wie das Versetzen des Cursors auf Adresse (0,0): sie ist die obere linke Ecke des Bildschirms, nicht des Speichers. (Daher kann die Sequenz \EH auf HP-Terminals nicht für das Versetzen des Cursors in die Ausgangsposition verwendet werden.)

Bereiche löschen

Wenn das Terminal von der aktuellen Position bis zum Ende der Zeile löschen kann, wobei der Cursor an seiner ursprünglichen Stelle verbleibt, wird dies durch el angegeben. Wenn das Terminal von der aktuellen Position bis zum Ende des Bildschirms löschen kann, wird dies mit ed angegeben. Ed ist nur ab der ersten Position einer Zeile definiert. (Es läßt sich also simulieren durch ein Kommando zum Löschen einer größeren Anzahl Zeilen, wenn die echte Funktion ed nicht vorhanden ist.)

Zeile einfügen/löschen

Wenn das Terminal eine neue Leerzeile über der Zeile, in der der Cursor steht, öffnen kann, wird dies mit il1 angegeben. Dies ist nur von der ersten Position einer Zeile aus möglich. Der Cursor muß dann in der neuen Leerzeile stehen. Wenn das Terminal die Zeile, in der der Cursor steht, löschen kann, wird dies mit dl1 angegeben. Dies ist nur von der ersten Position in der zu löschenden Zeile aus möglich. Varianten von il1 und dl1 sind il und dl; diese besitzen einen einzelnen Parameter und fügen eine entsprechende Anzahl Zeilen ein bzw. löschen sie. Wenn das Terminal einen einstellbaren Rollbereich besitzt (wie der

Anhang 3: Dateiformate

vt100), kann das Kommando zum Einstellen dieses Bereichs mit der Funktionsmöglichkeit `csr` beschrieben werden; `csr` besitzt zwei Parameter: die oberste und die unterste Zeile des Rollbereichs. Die Cursorposition ist nach diesem Kommando leider nicht definiert. Man kann mit diesem Kommando die Wirkung des Einfügens bzw. Löschsens einer Zeile erzielen – die Kommandos `sc` und `rc` (`save` bzw. `restore cursor`) sind ebenfalls nützlich. Auf vielen Terminals kann man mit `ri` bzw. `ind` Zeilen am oberen bzw. unteren Rand des Bildschirms einfügen, ohne eine echte Operation zum Einfügen/Löschen von Zeilen vorzunehmen, und diese Form ist häufig schneller, sogar auf Terminals, die diese Funktionen besitzen.

Wenn das Terminal ein Fenster als Teilbereich des Speichers definieren kann, auf das alle Kommandos wirken, muß dies als parametrisierte Zeichenkette `wind` angegeben werden. Die vier Parameter sind die erste und letzte Zeile im Speicher sowie die erste und letzte Spalte im Speicher in der genannten Reihenfolge.

Damit das Terminal den Bildschirmspeicherinhalt nach oben hin beibehalten kann, ist die Funktionsmöglichkeit `da` anzugeben; soll das Terminal den Bildschirmspeicherinhalt nach unten hin beibehalten, ist die Funktionsmöglichkeit `db` anzugeben. Dies bedeutet, daß beim Löschen einer Zeile oder beim Scrollen nicht-leere Zeilen von unten nach oben und daß beim Rückwärtsscrollen mit `ri` nicht-leere Zeilen nach unten gebracht werden können.

Zeichen einfügen/löschen

Bei den intelligenten Terminals sind hinsichtlich des Einfügens/Löschsens von Zeichen zwei Grundtypen zu unterscheiden, die in `terminfo` beschrieben werden können. Bei der häufigsten Form des Einfügens/Löschsens von Zeichen werden nur die Zeichen in der aktuellen Zeile betroffen und werden Zeichen unverändert und starr über das Ende der Zeile hinausgeschoben. Andere Terminals, z. B. der `Concept 100` und der `Perkin Elmer Owl`, unterscheiden zwischen „`typed`“ und „`untyped`“ Leerzeichen und verschieben nach einem Einfügen oder Löschen nur bis zu einem „`untyped`“ Leerzeichen auf dem Bildschirm, das entweder wegfällt oder zu zwei „`untyped`“ Leerzeichen expandiert wird. Welchen Terminaltyp Sie haben, können Sie feststellen, indem Sie den

Anhang 3: Dateiformate

Bildschirm löschen und dann durch Cursorbewegungen getrennte Textteile schreiben. Schreiben Sie abc def, wobei zwischen dem abc und dem def lokale Cursorbewegungen (keine Leerzeichen) eingegeben werden. Dann stellen Sie den Cursor vor das abc und stellen das Terminal auf den Einfügemodus ein. Wenn Sie jetzt weitere Zeichen eingeben und der Rest der Zeile unverändert starr verschoben wird und Zeichen am Zeilenende herausfallen, unterscheidet Ihr Terminal nicht zwischen Leerzeichen und „untyped“ Positionen. Wenn das abc zum def hinüberschoben wird und diese Zeichen dann beim weiteren Einfügen zusammen auf die nächste Zeile umgebrochen werden, besitzen Sie den anderen Terminaltyp, und Sie müssen die Funktionsmöglichkeit in angeben, wobei in für „insert null“ steht.

Terminfo kann sowohl Terminals beschreiben, die einen Einfügemodus besitzen, als auch solche Terminals, die eine einfache Kommandofolge absetzen, um eine Leerstelle in der aktuellen Zeile zu öffnen. Die Sequenz, mit der in den Einfügemodus umgeschaltet wird, wird mit smir angeben. Die Sequenz, mit der der Einfügemodus verlassen wird, wird mit rmir angeben. Jetzt geben Sie mit ich1 die Sequenz an, die eventuell unmittelbar vor der Übergabe des einzufügenden Zeichens abgesetzt werden muß. Die meisten Terminals mit einem echten Einfügemodus haben kein ich1; bei Terminals, die eine Sequenz zum Öffnen einer Bildschirmposition absetzen, muß ich1 hier angegeben werden. (Wenn Ihr Terminal über beide Betriebsarten verfügt, ist der Einfügemodus dem ich1 in der Regel vorzuziehen. Nicht beide Werte angeben, es sei denn, bei dem Terminal müssen tatsächlich beide zusammen verwendet werden.) Wenn nach dem Einfügen eine Zeitverzögerung erforderlich ist, muß diese in Anzahl Millisekunden mit ip (eine Zeichenkettenangabe) angegeben werden. Jede andere Sequenz, die möglicherweise nach dem Einfügen eines einzelnen Zeichens übergeben werden muß, kann auch bei ip angegeben werden. Wenn Ihr Terminal sowohl in einen 'Einfügemodus' gebracht als auch jedem eingefügten Zeichen ein spezieller Code vorangestellt werden muß, können smir/rmir und ich1 angegeben werden, und beide werden dann auch benutzt. Wird die Funktionsmöglichkeit ich1 mit einem Parameter n angegeben, so bewirkt dies eine n-malige Wiederholung der Wirkung von ich1.

Anhang 3: Dateiformate

Wenn man im Einfügemodus arbeitet, muß man hin und wieder Zeichen in der selben Zeile löschen (z. B. wenn hinter der Einfügeposition ein Tab steht). Wenn auf Ihrem Terminal im Einfügemodus Cursorbewegungen möglich sind, können Sie die Funktionsmöglichkeit `nir` angeben, um das Einfügen in diesem Fall zu beschleunigen. Wird `nir` weglassen, so wirkt dies nur auf die Geschwindigkeit im Einfügemodus. Bei einigen Terminals (insbesondere denen von Datamedia) darf `nir` wegen des dort implementierten Einfügemodus nicht angegeben werden.

Schließlich können noch folgende Funktionsmöglichkeiten angegeben werden: Das Löschen eines einzelnen Zeichens wird durch `dch1` angegeben, das Löschen von `n` Zeichen wird durch `dch` mit dem einzelnen Parameter `n` angegeben, das Aktivieren des Löschmodus wird durch `smdc` und das Verlassen des Löschmodus durch `rmdc` angegeben (wobei der Löschmodus die Betriebsart ist, in die das Terminal versetzt werden muß, damit `dch1` funktioniert).

Ein Kommando zum Löschen von `n` Zeichen (gleichbedeutend mit der Ausgabe von `n` Leerzeichen ohne Cursorbewegung) kann durch `ech` mit einem einzelnen Parameter angegeben werden.

Hervorhebungen und Unterstreichung

Wenn Ihr Terminal über ein oder mehrere Anzeigeattribute verfügt, können diese auf unterschiedliche Weise dargestellt werden. Sie sollten als Hervorhebungsmodus eine Anzeigeform auswählen, bei der Fehlermeldungen und andere Elemente, auf die man aufmerksam gemacht werden soll, in einem guten, kontrastreichen und augenfreundlichen Format hervorgehoben werden. (Wenn Sie die Wahl haben, ist invertierte Darstellung kombiniert mit halbhell oder invertiert allein vorteilhaft.) Die Sequenzen zum Aktivieren und Beenden des Hervorhebungsmodus werden als `smso` bzw. `rmso` angegeben. Wenn der Code, mit dem in den Hervorhebungsmodus umgeschaltet oder dieser beendet wird, ein oder zwei Leerzeichen auf dem Bildschirm verursacht, wie beispielsweise beim TVI 912 und Teleray 1061, sollte mit `xmc` angegeben werden, wie viele Leerzeichen entstehen.

Die Codes zum Beginnen und Beenden einer Unterstreichung werden mit `smul` bzw. `rmul` angegeben. Wenn das Terminal über einen Code

Anhang 3: Dateiformate

verfügt, mit dem das aktuelle Zeichen unterstrichen und der Cursor um eine Stelle nach rechts bewegt wird, wie z. B. das Microterm Mime, kann dies mit `uc` angegeben werden.

Weitere Funktionsmöglichkeiten zum Aktivieren verschiedener Hervorhebungsformen sind `blink` (blinkend), `bold` (fett oder besonders hell), `dim` (halbhell), `invis` (dunkelgesteuert oder unsichtbarer Text), `prot` (geschützt), `rev` (invertiert), `sgr0` (alle Attributarten deaktivieren), `smacs` (Alternativzeichensatz-Modus starten) und `rmacs` (Alternativzeichensatz-Modus verlassen). Wird eine dieser Anzeigeformen einzeln aktiviert, so werden eventuell andere Anzeigeformen beendet.

Wenn es eine Sequenz gibt, mit der beliebige Kombinationen von Modus gesetzt werden können, sollte diese mit `sgr` (set attributes) angegeben werden, wobei neun Parameter spezifiziert werden. Jeder Parameter ist entweder 0 oder 1, d. h. das entsprechende Attribut ist aktiviert bzw. ausgeschaltet. Die neun Parameter sind in dieser Reihenfolge: hervorgehoben, unterstrichen, invertiert, blinkend, halbhell, fett, dunkelgesteuert, geschützt, Alternativzeichensatz. Es müssen nicht alle Modus von `sgr` unterstützt werden, sondern nur diejenigen, für die entsprechende einzelne Attributkommandos vorhanden sind.

Wenn es auf dem Terminal möglich ist, den Bildschirm blinken zu lassen, um unhörbar auf einen Fehler hinzuweisen (als Ersatz für ein akustisches Signal), kann dies mit `flash` angegeben werden; der Cursor darf dabei nicht bewegt werden.

Wenn der Cursor dann, wenn er sich nicht in der untersten Zeile befindet, stärker als normal hervorgehoben werden muß (wobei beispielsweise ein nicht-blinkender Unterstrich in einen leichter zu findenden Block oder einen blinkenden Unterstrich umgewandelt wird), wird diese Sequenz mit `cvvis` angegeben. Wenn es möglich ist, den Cursor völlig unsichtbar zu machen, wird dies mit `cvis` angegeben. Die Funktionsmöglichkeit `cnorm` sollte angegeben werden, die die Wirkung dieser beiden Modus rückgängig macht.

Muß das Terminal sich in einem speziellen Modus befinden, wenn ein Programm ausgeführt wird, das diese Funktionsmöglichkeiten nutzt, können die Codes zum Aktivieren und Verlassen dieses Modus als `smcup` und `rncup` angegeben werden. Dies trifft beispielsweise bei

Anhang 3: Dateiformate

Terminals wie dem Concept zu, die mehr als eine Seite Bildschirmspeicher besitzen. Wenn das Terminal nur speicherbezogene Cursoradressierung und keine bildschirmbezogene Cursoradressierung besitzt, muß ein Fenster von der Größe eines Bildschirms im Terminal fixiert werden, damit die Cursoradressierung einwandfrei funktioniert. Dies wird auch bei dem TEKTRONIX 4025 verwendet, wo smcup das Kommandozeichen auf jenes Zeichen setzt, das von terminfo verwendet wird. Wenn Ihr Terminal unterstrichene Zeichen einwandfrei erzeugt (ohne daß spezielle Codes erforderlich sind), obwohl es nicht überschreiben kann, müssen Sie die Funktionsmöglichkeit ul angeben. Wenn Überschreibungen (Overstrikes) mit einem Leerzeichen löschar sind, muß dies mit eo angegeben werden.

Tastatur

Wenn das Terminal eine Tastatur besitzt, die die Codes dann absetzt, wenn die Tasten gedrückt werden, kann diese Information angegeben werden. Es ist jedoch nicht möglich, Terminals zu behandeln, bei denen die Tastatur nur im lokalen Modus arbeitet (dies gilt beispielsweise für die nicht geschifteten Tasten bei der HP 2621). Wenn die Tastatur auf Sendebetrieb und Nicht-Sendebetrieb eingestellt werden kann, geben Sie diese Codes durch smkx und rmkx an. Andernfalls wird davon ausgegangen, daß die Tastatur immer sendet. Die von den Tasten Linkspfeil, Rechtspfeil, Aufwärtspfeil, Abwärtspfeil und „Cursor in Ausgangsstellung“ abgesetzten Codes werden mit kcup1, kcuf1, kcuu1, kcud1 bzw. khome angegeben. Wenn Funktionstasten wie f0, f1, ..., f10 vorhanden sind, können die davon abgesetzten Codes mit kf0, kf1 usw. und die Stufen (labels) mit lf0, lf1, ..., lf10 angegeben werden. Die von bestimmten anderen Spezialtasten ausgesendeten Codes werden wie folgt angegeben: kll (Ausgangsstellung unten), kbs (Rückschritt), ktbc (alle Tabs löschen), kctab (Tabstop in dieser Spalte löschen), kclr (Bildschirm löschen oder Erase-Taste), kdch1 (Zeichen löschen), kdl1 (Zeile löschen), krmir (Einfügemodus beenden), kel (Löschen bis Zeilenende), ked (Löschen bis Bildschirmende), kich1 (Zeichen einfügen oder Einfügemodus aktivieren), kil1 (Zeile einfügen), knp (nächste Seite), kpp (vorgehende Seite), kind (vorwärts/abwärts rollen), kri (rückwärts/aufwärts rollen), khts (Tabstop in dieser Spalte setzen). Wenn darüber hinaus die Tastatur über ein Tastenfeld von drei mal drei Tasten einschließlich der vier Pfeiltasten verfügt, können die anderen fünf

Anhang 3: Dateiformate

Tasten mit ka1, ka3, kb2, kc1 und kc3 angegeben werden. Diese Tasten sind nützlich, wenn die Funktionen eines richtungsorientierten Cursorsteuertastenfeldes mit drei mal drei Steuertasten benötigt werden.

Tabs und Initialisierung

Wenn das Terminal über Hardware-Tabs verfügt, kann das Kommando für den Sprung zum nächsten Tabstop mit ht (meist Control-I) angegeben werden. Ein Rückwärtstab-Kommando, das den Cursor nach links zum nächsten Tabstop bringt, kann mit cbt angegeben werden. Grundsätzlich gilt, daß dann, wenn in den Teletype-Betriebsarten angegeben ist, daß der Rechner Tabs expandiert statt sie an das Terminal zu senden, die Programme nicht mit ht oder cbt arbeiten sollten, auch wenn diese vorhanden sind, da der Benutzer die Tabstops möglicherweise nicht sachgemäß gesetzt hat. Wenn das Terminal über Hardware-Tabs verfügt, die beim Einschalten anfangs in Form eines Tabgitters mit einem Abstand von n Stellen gesetzt sind, gibt der numerische Parameter den Abstand zwischen den Tabs in Anzahl Stellen an. Aufgrund dieser Angabe stellt das tset-Kommando in der Regel fest, ob der Modus für die Expandierung von Hardware-Tabs gesetzt werden muß und ob die Tabstops zu setzen sind. Wenn das Terminal über Tabstops verfügt, die in einem nicht-flüchtigen Speicher gesichert werden, kann in der terminfo-Beschreibung davon ausgegangen werden, daß sie sachgemäß gesetzt sind.

Weitere Funktionsmöglichkeiten sind is1, is2 und is3, d. h. Initialisierungs-Strings für das Terminal, iprog, d. h. der Pfadname eines Programms, das zur Initialisierung des Terminals auszuführen ist, und if, d. h. der Name einer Datei, die lange Initialisierungs-Strings enthält. Diese Strings müssen das Terminal auf Betriebsarten einstellen, die mit dem Rest der terminfo-Beschreibung verträglich sind. Sie werden in der Regel immer dann, wenn der Benutzer sich anmeldet, vom tset-Programm an das Terminal gesendet. Sie werden in der folgenden Reihenfolge ausgegeben: is1; is2; Tabs mit tbc und hts setzen; if; Ausführung des Programms iprog; und zuletzt is3. Der größte Teil der Initialisierung erfolgt mit is2. Spezielle Terminal-Betriebsarten können ohne Verdoppelung der Strings eingerichtet werden, wenn man die in jedem Fall geltenden Sequenzen in is2 und die Sonderfälle in is1 und is3 codiert.

Anhang 3: Dateiformate

Ein Sequenz-Paar, das eine stärker wirkende Rücksetzung aus einem völlig unbekanntem Zustand bewirkt, kann analog mit rs1, rs2, rf und rs3 angegeben werden, was analog zu is2 und if ist. Diese Strings werden vom reset-Programm ausgegeben, das dann benutzt wird, wenn das Terminal sich in einem ausweglosen Zustand befindet. Kommandos werden normalerweise nur dann in rs2 und rf untergebracht, wenn sie lästige Effekte auf dem Bildschirm hervorrufen und beim Anmelden nicht notwendig sind. Das Kommando, mit dem der vt100 auf 80 Spalten eingestellt wird, wäre beispielsweise normalerweise in is2 enthalten, bewirkt aber einen lästigen Störeffekt auf dem Bildschirm und ist normalerweise nicht erforderlich, da das Terminal in der Regel bereits auf 80 Spalten eingestellt ist.

Wenn es Kommandos zum Setzen und Löschen von Tabstops gibt, können diese mit tbc (alle Tabstops löschen) und hts (einen Tabstop in der aktuellen Spalte jeder Zeile setzen) angegeben werden. Wenn zum Setzen der Tabs eine kompliziertere Sequenz erforderlich ist als hierdurch beschrieben werden kann, kann diese Sequenz in is2 oder if untergebracht werden.

Verzögerungen

Bestimmte Funktionsmöglichkeiten steuern Zeitverzögerungen im Teletype-Treiber. Diese werden hauptsächlich für Hardcopy-Terminals benötigt, und damit stellt das tset-Programm die Teletype-Betriebsarten sachgerecht ein. Werden in die Funktionsmöglichkeiten cr, ind, cub1, ff und tab Verzögerungen eingebettet, so bewirken diese, daß die entsprechenden Verzögerungsbits im Teletype-Treiber gesetzt werden. Wenn pb (Padding-Baudrate) angegeben ist, können diese Werte bei Baudraten unterhalb des Wertes von pb ignoriert werden.

Verschiedenes

Wenn für das Terminal ein anderes Füllzeichen (Padding-Zeichen) als das Nullzeichen (Null) erforderlich ist, kann dies mit pad angegeben werden. Nur das erste Zeichen der pad-Zeichenkette wird berücksichtigt.

Anhang 3: Dateiformate

Wenn das Terminal eine zusätzliche Statuszeile besitzt, die normalerweise nicht von der Software benutzt wird, kann man diese Tatsache angeben. Wenn die Statuszeile als eine zusätzliche Zeile unterhalb der untersten Zeile angesehen wird, in die man mit normaler Cursoradressierung gelangen kann (wie beispielsweise die 24. Zeile beim vt100, der auf einen Rollbereich von 23 Zeilen eingestellt ist), muß die Funktionsmöglichkeit `hs` angegeben werden. Spezielle Strings, mit denen man an den Anfang der Statuszeile springt und aus der Statuszeile zurückkehrt, können mit `tsl` und `fsl` angegeben werden. (`fsl` muß den Cursor an der selben Stelle lassen, an der er sich vor `tsl` befand. Wenn nötig, können die Strings `sc` und `rc` in `tsl` und `fsl` einbezogen werden, um diese Wirkung zu erzielen.) Der Parameter `tsl` besitzt einen Parameter, und zwar die Spaltennummer in der Statuszeile, in die der Cursor gestellt werden soll. Wenn Escape-Folgen und andere spezielle Kommandos wie etwa `tab` in der Statuszeile wirksam sind, kann der Merker `eslok` angegeben werden. Ein String, der die Statuszeile deaktiviert (oder ihren Inhalt sonstwie löscht), muß mit `dsl` angegeben werden. Wenn das Terminal Kommandos besitzt, mit denen die Cursorposition gespeichert und wiederhergestellt werden kann, müssen diese mit `sc` und `rc` angegeben werden. Es wird davon ausgegangen, daß die Statuszeile normalerweise genauso lang ist wie die anderen Zeilen auf dem Bildschirm, z. B. `cols`. Wenn die Statuszeile eine abweichende Zeilenlänge hat (z. B. weil es bei dem Terminal nicht möglich ist, eine vollständige Zeile zu laden), kann die Breite in Spalten mit dem numerischen Parameter `ws` angegeben werden.

Wenn das Terminal Halbzeilenschaltungen nach oben und unten durchführen kann, kann dies mit `hu` (1/2 hoch) bzw. `hd` (1/2 tief) angegeben werden. Dies ist in erster Linie sinnvoll für hochgestellte und tiefgestellte Zeichen auf Hardcopy-Terminals. Wenn ein Hardcopy-Terminal einen Seitenvorschub zur nächsten Seite (Formularvorschub) ausführen kann, muß dies mit `ff` (normalerweise Control-L) angegeben werden.

Anhang 3: Dateiformate

Wenn es ein Kommando gibt, mit dem ein bestimmtes Zeichen mit einem bestimmten Faktor wiederholt werden kann (um beim Übertragen einer größeren Anzahl identischer Zeichen Zeit zu sparen), kann dies mit der parametrisierten Zeichenkette rep angegeben werden. Der erste Parameter ist das zu wiederholende Zeichen, der zweite Parameter ist der Wiederholungsfaktor.

Terminalbeschreibungen, die sich nicht auf eine bestimmte Art eines bekannten Terminals beziehen, z. B. switch, dialup, patch und network, sollten die Funktionsmöglichkeit gn (generisch) enthalten, damit Programme melden können, daß sie nicht wissen, wie sie mit dem Terminal kommunizieren sollen. (Diese Funktionsmöglichkeit gilt nicht für virtuelle Terminalbeschreibungen, für die die Escape-Folgen bekannt sind.)

Wenn das Terminal die Datenflußkontrolle mit dem xon/xoff-Protokoll bewirkt, muß xon angegeben werden. Angaben über die Verzögerung (padding) sollten gemacht werden, damit die Programme eine bessere Entscheidung bezüglich der Kosten treffen können, aber die Füllzeichen selbst werden nicht übertragen.

Wenn das Terminal eine „Meta-Taste“ besitzt, die als Umschalttaste wirkt und das 8. Bit jedes gesendeten Zeichens setzt, kann dieser Umstand mit km angegeben werden. Andernfalls nimmt die Software an, daß das 8. Bit das Paritätsbit ist und löscht dies im Normalfall. Wenn Strings vorhanden sind, mit denen dieser „Meta-Modus“ ein- und ausgeschaltet werden kann, können diese mit smm und rmm angegeben werden.

Wenn das Terminal mehr Zeilen im Speicher hat als auf einmal auf den Bildschirm passen, kann die Anzahl der Zeilen im Speicher mit lm angegeben werden. Der Wert lm#0 gibt an, daß die Anzahl der Zeilen nicht fest ist, daß der Speicher aber größer als ein Bildschirminhalt ist.

Wenn das Terminal vom UNIX Virtual Terminal Protocol unterstützt wird, kann die Terminalnummer mit vt angegeben werden.

Strings zum Programmieren von Funktionstasten können mit pfkey, pfloc und pfx angegeben werden. Jeder dieser Strings besitzt zwei Parameter: die zu programmierende Funktionstastennummer (von 0 bis

© „Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Mitteilung ihres Inhalts nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zu Schadenersatz. Alle Rechte für den Fall der Patentierung oder Gebrauchsmustereintragung vorbehalten.“

A3

Anhang 3: Dateiformate

10) und die Zeichenkette, mit der diese Taste zu programmieren ist. Mit Funktionstastennummern außerhalb dieses Bereichs können unter Umständen nicht definierte Tasten terminalabhängig programmiert werden. Der Unterschied zwischen diesen Funktionsmöglichkeiten ist folgender: `pfkey` bewirkt, daß das Betätigen der betreffenden Taste die selbe Wirkung hat als wenn der Benutzer die entsprechende Zeichenfolge eingibt; `pfloc` bewirkt, daß die Zeichenfolge vom Terminal im Lokalmodus ausgeführt wird; `pxf` bewirkt, daß die Zeichenfolge an den Computer übertragen wird.

Spezialfälle

Für Hazeltine-Terminals, auf denen das Zeichen `~` nicht angezeigt werden kann, muß `hz` angegeben werden.

Für Terminals, die einen Zeilenvorschub unmittelbar nach einem Zeilenwechsel gemäß `am` ignorieren, wie der `vt100`, muß `xenl` angegeben werden.

Wenn `el` notwendig ist, um eine Hervorhebung aufzuheben (statt einfach normalen Text darüberzuschreiben), muß `xhp` angegeben werden.

Für die `Beehive Superbee`, die die `Escape`- oder `Control-C`-Zeichen nicht korrekt übergeben kann, wird `xsb` angegeben. Dies bedeutet, daß die Taste `f1` für `Escape` und `f2` für `Control-C` verwendet wird. (Dieses Problem gibt es nur bei bestimmten `Superbees` und hängt vom `ROM` ab.)

Weitere spezielle Terminalprobleme lassen sich durch Hinzunahme weiterer Funktionsmöglichkeiten der Form `xx` lösen.

Anhang 3: Dateiformate

Ähnliche Terminals

Wenn man zwei sehr ähnliche Terminals hat, kann man eins davon so definieren, daß es – bis auf die entsprechenden Ausnahmen – gleich dem anderen ist. Die Zeichenketten-Funktionsmöglichkeit use kann mit dem Namen des ähnlichen Terminals angegeben werden. Die vor use angegebenen Funktionsmöglichkeiten setzen die in dem von use herangezogenen Terminaltyp angegebenen Funktionsmöglichkeiten außer Kraft. Man kann eine Funktionsmöglichkeit löschen, indem man links von der Definition der Funktionsmöglichkeit ein xx@ angibt, wobei xx die Funktionsmöglichkeit ist.

Der Eintrag

2621-nl, smkx@, rmkx@, use=2621,

definiert beispielsweise ein Terminal 2621-nl, das die Funktionsmöglichkeiten smkx und rmkx nicht besitzt und daher die Funktionstastenschriftungen im visuellen Modus nicht einschaltet. Dies ist sinnvoll bei unterschiedlichen Betriebsarten eines Terminals oder um unterschiedlichen Wünschen der Benutzer entgegenzukommen.

DATEIEN:

/usr/lib/terminfo/?/* Dateien mit Terminalbeschreibungen

Anhang 3: Dateiformate

utmp, wtmp – Format der Einträge in utmp und wtmp

SYNTAX:

```
#include <sys/types.h>
#include <utmp.h>
```

BESCHREIBUNG:

Diese Dateien, die Benutzer- und Abrechnungsinformationen für Kommandos wie who, write und login enthalten, beinhalten für jeden angemeldeten Benutzer eine Struktur von folgendem in <utmp.h> definiertem Typ:

```
#define  UTMP_FILE  "/etc/utmp"
#define  WTMP_FILE  "/etc/wtmp"
#define  ut_name    ut_user

struct utmp
{
    char  ut_user [8];          /* User login name */
    char  ut_id [4];           /* /etc/inittab id (usually line #) */
    char  ut_line [12];        /* device name (console, lnxx) */
    short ut_pid;              /* process id */
    short ut_type;             /* type of entry */
    struct exit_status
    {
        short e_termination; /* Process termination status */
        short e_exit;         /* Process exit status */
    } ut_exit;                 /* The exit status of a process
                               * marked as DEAD_PROCESS. */
    time_t ut_time;           /* time entry was made */
};
```

Anhang 3: Dateiformate

```

/* Definitions for ut_type */
#define EMPTY      0
#define RUN_LVL    1
#define BOOT_TIME  2
#define OLD_TIME   3
#define NEW_TIME   4
#define INIT_PROCESS 5 /* Process spawned by "init" */
#define LOGIN_PROCESS 6 /* A "getty" process waiting for login */
#define USER_PROCESS 7 /* A user process */
#define DEAD_PROCESS 8
#define ACCOUNTING 9
#define UTMATYPE   ACCOUNTING /* Largest legal value of uttype */

/* Special strings or formats used in the "ut_line" field when */
/* accounting for something other than a process */
/* No string for the ut_line field can be more than 11 chars + */
/* a NULL in length */
#define RUNLVL_MSG "run-level %c"
#define BOOT_MSG  "system boot"
#define OTIME_MSG "old time"
#define NTIME_MSG "new time"

```

DATEIEN:

/usr/include/utmp.h

/etc/utmp

/etc/wtmp

Anhang 4: Stichwortverzeichnis

anlegen, FIFO-Datei (mknod) A1
anlegen, Named-Pipe (mknod) A1
AP 2.1.2
Applikations-Prozessor 2.1.2
Archivbibliotheken 3.4.3.2
aufbauen, Dateisystem (mkfs) A1
Auftrags-ID 5.2.1
Ausgabeanforderung 5.1
aushängen, Dateisystem (umount) A1
Auswahlmenü 3.3.2, 3.4
Auswahlschalter 3.2
auswerten, Fehler (errpt) A1
auswerten, Plattenzugriffe (sadb) A1
auswerten, Systemaktivitäten (sar) A1

Baud-Rate 5.6.2
bcheckrc (brc) A1
Bedienungspanel 3.2
beenden, Fehlerprotokollierungs-Dämon (errstop) A1
beenden, Verarbeitung (shutdown) A1
Benutzerkommandos 5.2.1
Benutzerstatistik 4.6
Benutzerumgebung (profile) A3
Betriebssystem-Profiler (prf) A2
Betriebssystem-Profiler (profiler) A1
Betriebssystemteile 3.3.1
Betriebssystemversion 3.3.1
bin 3.4.3.1, 3.4.3.2
Bit-Slice-Prozessor 2.2.3
boot (inittab) A3
Boot-Phasen 3.1
Boot-Verfahren 3
Boot1 3.1
Boot2 3.1
bootwait (inittab) A3
brc A1
Burst-Error-Prozessor 2.2.3

Anhang 4: Stichwortverzeichnis

C-inkludate-Dateien 3.4.3.2
 Cache-Streamer-Tape 2.3
 cancel 5.2.1
 chargefee (acctsh) A1
 checkall A1
 checklist A3
 chroot 4.3, A1
 ckpacct (acctsh) A1
 clri A1
 core A3
 Core-Image Datei (core) A3
 Core-Images untersuchen (crash) A1
 cpio A3
 cpio-Archiv (cpio) A3
 cpio-Format 4.4.2
 cpset A1
 crash 3.3.4, A1
 cron 4.5.4, 4.6, A1
 cronlog 4.5.4
 crontab 4.5.4, 4.6

Daemons 4.1
 Dataproducts-Longline 2.1.6.2
 Dataproducts-Shortline 2.1.6.2
 Dateien rekonstruieren (frec) A1
 Dateisystem aufbauen (mkfs) A1
 Dateisystem aushängen (umount) A1
 Dateisystem einhängen (mount) A1
 Dateisystem-Debugger (fsdb) A1
 Dateisysteme reorganisieren (dcopy) A1
 Datenschutzmechanismen 4.3
 Datensicherung, inkrementelle (finc) A1
 DTEX 2.1.6.1
 dcopy A1
 dev 3.4.3.1
 devnm A1
 df 4.5.2, A1
 dfsck A1
 disable 5.2.1

Anhang 4: Stichwortverzeichnis

Disk/Tape-Prozessor 2.2.3
diskusg A1
DMA 2.2.3
dodisk (acctsh) A1
Druckaufträge freigeben (accept) A1
Druckaufträge sperren (reject) A1
Druckaufträge verschieben (lpmove) A1
Drucker 2.3, 5.1
Drucker löschen 5.4.5
Drucker, serielle 2.3
Druckerkonfiguration 5.4.1
Druckername 5.4.1.1
Druckerspezifizierung 5.4.2
DTP 2.2.3
du 4.5.2
Dual-Port-Anschluß 2.1.6

ECC 2.1.3
einhängen, Dateisystem (mount) A1
einrichten, mount-Tabelle (setmnt) A1
einstellen, Geschwindigkeit (getty) A1
einstellen, Leitungsprozedur (getty) A1
einstellen, Terminalmodi (getty) A1
einstellen, Terminaltyp (getty) A1
enable 5.2.1
ermitteln, Gerätenamen (devnm) A1
err A2
errdead A1
errdemon A1
errfile 4.5.4, A3
Error-Correction-Control 2.1.3
errpt A1
errstop A1
Ersatzspur-Tabelle 3.4.6
etc 3.4.3.1
extrahieren, Fehlersätze (errdead) A1

Anhang 4: Stichwortverzeichnis

Fehler auswerten (errpt) A1
 Fehlermeldungen 5.5
 Fehlerprotokolldateien (errfile) A3
 Fehlerprotokollierungs-Dämon (errdemon) A1
 Fehlerprotokollierungs-Dämon beenden (errstop) A1
 Fehlersätze extrahieren (errdead) A1
 Fernbetreuung 2.1.6.1
 ff A1
 FIFO-Datei anlegen (mknod) A1
 filesave A1
 finc A1
 find 4.5.2
 find -cpio 4.4.2
 First Level Boot 3.1
 Format, I-Knoten (inode) A3
 frec A1
 freie Plattenblöcke melden (df) A1
 freigeben, Druckaufträge (accept) A1
 fs A3
 fsck 4.4.1, A1
 fsdb A1
 fuser A1
 fwtmp A1
 Füllgrad 4.5.2

 Gerätedateien 5.1, 5.4.1.1
 Gerätenamen ermitteln (devnm) A1
 Gerätezuordnung 5.4.2
 Geschwindigkeit einstellen (getty) A1
 getty A1
 getty-Aufruf 5.7.1
 gettydefs A3
 group A3
 grpck A1
 Gruppendatei (group) A3
 Gruppendatei prüfen (grpck) A1
 Gruppenzuordnung 4.1

Anhang 4: Stichwortverzeichnis

Hardware-Cache 2.1.2
Hardware-Fehlerprotokolle 4.5.4
Hardware-Selbsttest 3.2
Hardware-Testprogramme 3.1
Hauptspeicher-Abbild (mem) A2
HIGH DENSITY 3.4.2
hog factor (acctcms) A1

I-Knoten löschen (clri) A1
I-Knoten-Format (inode) A3
I/O-Busse 2.1.6
I/O-Controller 2.1.5
I/O-Etage 2.1.6
IHSS 2.1.6.1
include 3.4.3.2
init A1
Init-Prozeß 3.4.1
initdefault (inittab) A3
initialisieren, Prozeßsteuerung (init) A1
inittab A3
inkrementelle Datensicherung (finc) A1
inode A3
Input/Output-Prozessor 2.1.5
install A1
Installation 5.3
Installationskommandos (install) A1
IOP 2.1.5
iop 3.3.1
issue A3

kcore minutes A1
kill 5.7.1
killall A1
Klasse 5.1
Klassennamen 5.4.1.2
kmem A2
Kommandozeilen generieren 5.6.1
Konfiguration 5.1
konfigurieren, Spool-System (lpadmin) A1

Anhang 4: Stichwortverzeichnis

Konsistenzprüfungen A1

labelit A1

Ladestufe 1 3.3

Ladestufe 2 3.3

lastlogin (acctsh) A1

LED-Anzeige 3.2

Leitungsprozedur einstellen (getty) A1

lib 3.4.3.1, 3.4.3.2

link A1

Log-Dateien 4.5.4, 5.5

Login-Sicherheitsmaßnahmen (security) A3

lost+found 3.4.3.1, 3.4.3.2

LOW DENSITY 3.4.2

lp 5.2.1

lpadmin 5.2.2, A1

lpmove 5.2.2, A1

lpsched 5.2.2, 5.5.1, A1

lpshut 5.2.2, 5.5.2, A1

löschen, I-Knoten (clri) A1

Magnetbandgeräte 2.3

Magnetplattenlaufwerke 2.3

mail 3.4.3.2

Mail-Dateien 3.4.3.2

Matrixdrucker 2.3

melden, freie Plattenblöcke (df) A1

mem A2

Memory-Management-Unit 2.1.2

MFC 2.1.6.2

mkfs A1

mknod A1

MMU 2.1.2

mnttab A3

Modellschnittstellen 5.4.1.1, 5.6

monacct (acctsh) A1

mount A1

mount-Tabelle einrichten (setmnt) A1

Multi-User-Betrieb 3.3.1

Anhang 4: Stichwortverzeichnis

Multicommand-Interface 2.2.3
Multifunktions-Controller 2.1.6.2
mmdir A1

Named-Pipe anlegen (mknod) A1
NCAG-DP 2.1.6.2
ncheck A1
newgrp 4.2
news 3.4.3.2, 4.5.1
Notstromversorgung 2.1.3
null A2
Null-Datei (null) A2
nulladm (acctsh) A1

off (inittab) A3
once (inittab) A3
ondemand (inittab) A3

Parallel-Schnittstelle 2.1.6.2
Parameter, nicht-optionale 5.4.1.1
Parameter, optionale 5.4.1.2
Parity-Prüfung 2.1.3
passwd A3
Paßwortdatei (passwd) A3
Paßwortdatei prüfen (pwck) A1
Peripheriegerät, physikalisches 5.1
Peripheriegeräte 2.3
Plattenabrechnungsdaten (diskusg) A1
Plattenbelegung 4.5.2
Plattenverwendung 4.5.2
Plattenzugriffe auswerten (sadb) A1
Point-to-Point-Kanäle 2.1.6.1
Positivdarstellung 2.3
powerfail (brc) A1
powerfail (inittab) A3
powerwait (inittab) A3
prctmp (acctsh) A1
prdaily (actsh) A1
prf A2

Anhang 4: Stichwortverzeichnis

prfdc (profiler) A1
 prfld (profiler) A1
 prfpr (profiler) A1
 prfsnap (profiler) A1
 prfstat (profiler) A1
 profile A3
 profiler A1
 Profiler, Betriebssystem (prf) A2
 Programm-Programm-Kommunikation 2.3
 Prozesse abbrechen (killall) A1
 Prozeßabrechnung (acctprc) A1
 Prozeßabrechnungsdateien (acct) A3
 Prozeßsteuerung initialisieren (init) A1
 prtacct (acctsh) A1
 prüfen, Gruppendatei (grpck) A1
 prüfen, Paßwortdatei (pwck) A1
 Pseudogerätetreiber (sxt) A2
 pub 3.4.3.2
 pwck A1

rc (brc) A1
 reject 5.2.2, A1
 rekonstruieren, Dateien (frec) A1
 release 3.4.3.2
 reorganisieren, Dateisysteme (dcopy) A1
 respawn (inittab) A3
 RS232C 2.1.6.1
 runacct (acctsh) A1
 Rundschreiben senden (wall) A1
 Rückgabecode 5.6.3

sa1 A1
 sa2 A1
 sadc A1
 sadp A1
 sar A1
 SAS-Protokoll 2.1.6.2
 SCHEDLOCK 5.5.1
 Scheduler 5.1, 5.5

Anhang 4: Stichwortverzeichnis

Scheduler starten 5.5.1
Scheduler stoppen 5.5.2
Schnittstelle, serielle 2.1.6.2
Schnittstellenkarten 2.1.6.1
Schnittstellenprogramm 5.1, 5.4.1.1, 5.5, 5.6
Schutzmechanismen 4.4.1
SCSI-Bus 2.2.3
Second Level Boot 3.1
security A3
senden, Rundschreiben (wall) A1
Set-Group-ID-Bit 4.3
Set-User-ID-Bit 4.3
Set-User-ID-Erlaubnis 5.3.1
Set-User-ID-Programme 4.3
setmnt A1
shutacct (acctsh) A1
shutdown A1
Sicherung, 8-wöchige 4.4.1
Sicherung, tägliche 4.4.1
Sicherung, wöchentliche 4.4.1
Sicherungskopie 4.4
Sicherungsmedium 4.4.1
Single-User-Modus 3.3.1
spc 3.3.1
Speicher-Dump (errdead) A1
Speicherabbild 3.3.4
Speicherplatz 4.5.2
sperrn, Druckaufträge (reject) A1
spool 3.4.3.2, 4.5.4
Spool-System 5
Spool-System konfigurieren (lpadmin) A1
Spooler-Dateien 5.2.2
Spooler-Kommandos 5.2.2
Spooler-Konfiguration 5.4
Spooler-Scheduler starten (lpsched) A1
Spooler-Scheduler stoppen (lpshut) A1
Spooler-Verzeichnis 5.3
SPP 2.1.4, 2.2.2
Standalone-Programme 3.3.2

Anhang 4: Stichwortverzeichnis

Standard-Ausgabegerät 5.1
 Standard-Benutzerumgebung 4.1
 Standard-Betriebssystemversion 3.3.1
 Standard-Systemdrucker 5.4.4
 starten, Spooler-Scheduler (lpsched) A1
 Startsmall 3.1
 startup (acctsh) A1
 Statuslogbuch 4.5.4
 Steckkarten 2.1.6.2
 stoppen, Spooler-Scheduler (lpshut) A1
 Streamer-Cassette 2.3
 su 4.3, 4.6
 Superuser-Paßwort 4.3
 Superuser 5.2.2
 sxt A2
 sysdef A1
 sysinit (inittab) A3
 System Peripherals Controller 3.3.1
 System-Peripherie-Prozessor 2.1.4, 2.2.2
 Systemaktivitäten auswerten (sar) A1
 Systemdefinition (sysdef) A1
 Systemdrucker 2.1.6.2
 Systeminitialisierung (brc) A1
 Systemkonsole 2.1.6.1
 Systemumgebung 4.1

 Tagesnachrichten 4.5.1
 tapesave A1
 tar 4.4.2
 TCC 2.1.6.1
 telinit A1
 term A3
 Terminal-Communication-Controller 2.1.6.1
 Terminal-Modi 5.6.2
 Terminalfunktionen (terminfo) A3
 Terminalmodi einstellen (getty) A1
 Terminalperipherie 2.3
 Terminals 2.3
 Terminalschnittstelle (termio) A2

Anhang 4: Stichwortverzeichnis

Terminaltyp einstellen (getty) A1
terminfo A3
terminfo-Compiler (tic) A1
termio A2
Testsoftware 3.2
tic A1
tmp 3.4.3.1, 3.4.3.2
tty A2
turnacct (acctsh) A1
tägliche Abrechnungen (runacct) A1

überwachen, uucp-Netz (uusub) A1
umbenennen, Verzeichnisse (mvdir) A1
umount A1
unix 3.3.1
UNIX to UNIX Copy 2.1.6.1
unlink A1
untersuchen, Core-Images (crash) A1
usr 3.4.3.1
utmp A3
uuclean A1
UUCP 2.1.6.1
uucp-Netz überwachen (uusub) A1
uusub A1

V.11/X.21 2.1.6.1
V.24/V.28 2.1.6.1
Verarbeitung beenden (shutdown) A1
verschieben, Druckaufträge (lpmove) A1
verschieben, Verzeichnisse (mvdir) A1
Verwaltungsdateien 4.1, 4.5.4
Verzeichnisse umbenennen (mvdir) A1
Verzeichnisse verkleinern 4.5.3
Verzeichnisse verschieben (mvdir) A1
volcopy A1

wait (inittab) A3
wall 4.5.1, A1
wechseln, Wurzelverzeichnis (chroot) A1

Anhang 4: Stichwortverzeichnis

whodo A1
Winchester-Festplatten 2.3
wtmp A3
wtmpfix (fwtmp) A1
Wurzelverzeichnis wechseln (chroot) A1
Wählanschluß 4.3

Zeilendrucker 2.3

