

**NIXDORF**  
COMPUTER

TARGON® /35

# **Systemadministration**



Bitte abtrennen und in  
die Tasche im Handbuckrücken  
einstecken.

**Systemliteratur**  
TARGON®/35

Bitte senden Sie Änderungen und Ergänzungen  
zu diesem Literaturteil an die folgende Anschrift.

Betreuende Nixdorf-Niederlassung (unbedingt  
angeben):

Name: \_\_\_\_\_  
in Firma: \_\_\_\_\_  
oder Bereich NCAG: \_\_\_\_\_

Verkehrsnummer:  
10251.00.1.93

Ausgabedatum der letzten Änderung  
lt. Organisationsblatt (unbedingt angeben): \_\_\_\_\_

Ihre Aufnahme in die Verteilerliste für den Änderungsdienst erfolgt nur, wenn Sie diese Karte einschicken.  
Änderungen erhalten Sie durch die betreuende Nixdorf-Niederlassung.



Postkarte

Nixdorf Computer AG  
Abt. ZSI  
Fürstenallee 7

D-4790 Paderborn  
West-Germany



Einleitung	1
Kurzbeschreibung Hardware TARGON /35	2
Console Operating System (COS)	3
COS-Frames für Systemadministratoren	4
Nachrichten in der Systemstatus-Zeile	5
COS-Reset und System-Restart	6
Installation des Betriebssystems	7
Systemstart TARGON /35	8
Das Zwei-Welten-Konzept der TARGON /35	9
Initialisierungsprogramme	10
Laufende Arbeiten der Systemverwaltung	11
Das Spoolsystem	12



**Anhang 1: I/O-Fehler-Codes**

**Anhang 2: Bandlaufwerk: Bedienung/Fehler-Codes**

**Anhang 3: Administrator-Kommandos**

**Anhang 4: Gerätedateien**

**Anhang 5: Dateiformate**

**Anhang 6: Stichwortverzeichnis**

**A1**  
**A2**  
**A3**  
**A4**  
**A5**  
**A6**





---

## 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.

Erstaufgabe:

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**

<b>1</b>	<b>Einleitung</b> .....	1-1
<b>2</b>	<b>Kurzbeschreibung Hardware TARGON /35</b> .....	2-1
2.1	Generelle Übersicht.....	2-1
2.1.1	Systemschrank.....	2-2
2.1.2	Peripherieschrank.....	2-3
2.2	Prozessoren.....	2-4
2.2.1	Applikationsprozessor.....	2-5
2.2.2	Gleitkommaprozessor.....	2-6
2.2.3	Intelligenter Terminal-Prozessor.....	2-6
2.2.4	Input/Output-Prozessor.....	2-7
2.2.5	System-Support-Prozessor.....	2-8
2.3	CPU-Etage.....	2-9
2.4	Speicherausstattung.....	2-11
2.5	Peripherie.....	2-12
<b>3</b>	<b>Console Operating System (COS)</b> .....	3-1
3.1	Diskettenlaufwerk.....	3-1
3.2	COS Frames.....	3-2
3.2.1	Systemstatus-Zeile.....	3-3
3.2.2	Auswahl der Frames.....	3-3
3.2.3	Funktionsauswahl innerhalb eines Frames.....	3-5
3.2.4	Cursor-Kontrolle und Dateneingabe.....	3-5
3.2.5	Der COS-Eingabepuffer.....	3-6
3.3	Systemnachrichten.....	3-6
<b>4</b>	<b>COS-Frames für Systemadministratoren</b> .....	4-1
4.1	Index-Frame.....	4-1
4.1.1	Ändern des COS-Schlüssels.....	4-2
4.2	Frame 1.....	4-3
4.2.1	Frame-Auswahl und CPU-Kontrolle in Frame 1.....	4-4
4.2.2	Features in Frame 1.....	4-5
4.2.3	Anzeige der Speicher in Frame 1.....	4-7
4.2.4	Anzeige der I/O-Geräte in Frame 1.....	4-7
4.2.5	Operatoreingaben in Frame 1.....	4-8
4.3	Frame 0.....	4-11
4.3.1	Terminalstatus-Zeile.....	4-11
4.3.2	Verlassen des Frame 0.....	4-12
4.3.3	Abbruch von Frame 0.....	4-12
4.3.4	Verwendung des COS-Schlüssels aus Programmen.....	4-12

---

## Inhaltsverzeichnis

---

<b>5</b>	<b>Nachrichten in der Systemstatus-Zeile</b> .....	5-1
5.1	Meldungen in Feld 1.....	5-2
5.2	Meldungen in Feld 2.....	5-2
5.3	Meldungen in Feld 3.....	5-3
5.4	Meldungen in Feld 4.....	5-4
5.5	Meldungen in Feld 5.....	5-4
5.6	Meldungen in Feld 6.....	5-5
5.7	Meldungen in Feld 7.....	5-5
5.8	Meldungen in Feld 8.....	5-6
5.9	Ausgabe der Systemaktivitäten.....	5-7
<b>6</b>	<b>COS-Reset und System-Restart</b> .....	6-1
6.1	Abläufe beim COS-Reset.....	6-1
6.2	Abläufe beim System-Restart.....	6-2
<b>7</b>	<b>Installation des Betriebssystems</b> .....	7-1
7.1	Auslieferung des Betriebssystems.....	7-1
7.2	Laden des Betriebssystems.....	7-1
7.2.1	Einspielen des Root-Dateisystems.....	7-2
7.2.2	Booten des Betriebssystems.....	7-5
7.2.3	Laden des Dateisystems /usr.....	7-7
7.2.4	Anpassung systemspezifischer Dateien.....	7-7
7.2.5	Anlegen von Sicherungskopien neuer Dateisysteme.....	7-8
7.3	Generieren eines neuen Kernels.....	7-9
7.4	Abschließende Arbeiten.....	7-9
<b>8</b>	<b>Systemstart TARGON /35</b> .....	8-1
8.1	Einschalten und Systemstart.....	8-1
8.2	Restart der TARGON /35.....	8-2
8.3	Boot-Programme.....	8-2
8.3.1	Autoboot des Betriebssystems.....	8-2
8.3.2	Manuelles Booten des Betriebssystems.....	8-3
8.4	Shutdown und Reboot.....	8-3
8.5	Systemfehlermeldungen.....	8-4
8.6	Verhalten bei Stromausfall.....	8-4
<b>9</b>	<b>Das Zwei-Welten-Konzept der TARGON /35</b> .....	9-1
9.1	Kommandos, Systemschnittstellen und Spezialdateien.....	9-1
9.2	Pipes zwischen att und ucb.....	9-2
9.3	Dateistrukturen.....	9-3

## Inhaltsverzeichnis

<b>10</b>	<b>Initialisierungsprogramme</b> .....	10-1
10.1	Terminal-, Login- und Accounting-Dateien.....	10-3
10.2	rc-Dateien.....	10-7
10.2.1	System V-init.....	10-7
10.2.2	4.2 BSD-init.....	10-8
10.3	Login-Dateien für Benutzer.....	10-9
10.3.1	Reparaturen in /etc/inittab und /etc/utmp.....	10-9
10.3.2	Reparaturen in wtmp.....	10-10
10.3.3	Getty-Prozesse.....	10-11
10.4	Login-Sicherheitsmaßnahmen.....	10-11
10.5	Einrichten der Umgebung für csh und tcsh.....	10-12
10.6	Wechseln der Initialisierungsprogramm-Versionen.....	10-13
10.6.1	Vorgehensweise beim Austausch des System V-Pakets.....	10-14
10.6.2	Vorgehensweise beim Austausch des 4.2 BSD-Pakets.....	10-15
<b>11</b>	<b>Laufende Arbeiten der Systemverwaltung</b> .....	<b>11-1</b>
11.1	Datenschutzmechanismen.....	11-1
11.2	Sicherung und Wiederherstellung von Daten.....	11-2
11.2.1	Schutzmechanismen für Benutzerdateien.....	11-2
11.2.2	Programme zur Sicherung von Dateisystemen.....	11-3
11.3	Verkleinern von Verzeichnissen.....	11-4
11.4	Reinigen der administrativen Log-Dateien.....	11-5
11.5	Kommunikation mit den Benutzern.....	11-6
11.6	Überwachung der Plattenverwendung.....	11-6
11.7	Definition der Standard-Benutzerumgebung (/etc/profile).....	11-8
11.8	Eintragen neuer Benutzer.....	11-8
11.9	Zeitplan des Systemverwalters.....	11-10
<b>12</b>	<b>Das Spoolsystem</b> .....	<b>12-1</b>
12.1	Konfiguration.....	12-1
12.2	Kommandos.....	12-2
12.2.1	Benutzerkommandos.....	12-2
12.2.2	Administrator Kommandos.....	12-3
12.3	Die Installation des Spoolsystems.....	12-4
12.3.1	Vorsichtsmaßnahmen.....	12-4
12.4	Die Konfiguration des Spoolers mit lpadmin.....	12-5
12.4.1	Druckerkonfiguration.....	12-5
12.4.1.1	Nicht-optionale Parameter.....	12-5
12.4.1.2	Optionale Parameter.....	12-6
12.4.1.3	Beispiele zur Druckerkonfigurierung.....	12-7

---

## Inhaltsverzeichnis

---

12.4.2	Änderung von Druckern oder Druckerklassen.....	12-7
12.4.3	Beispiele zur Änderung von Druckern.....	12-8
12.4.4	Einrichten des Standard-Systemdruckers.....	12-9
12.4.5	Löschen von Druckern oder Klassen.....	12-10
12.5	Der Scheduler.....	12-11
12.5.1	Starten des Schedulers durch Ipsched.....	12-11
12.5.2	Stoppen des Schedulers durch Ipshut.....	12-12
12.6	Schnittstellenprogramme für Drucker.....	12-13
12.6.1	Generierung von Kommandozeilen.....	12-14
12.6.2	Ausgabe mit Hilfe von Schnittstellenprogrammen.....	12-14
12.6.3	Rückgabecodes der Schnittstellenprogramme.....	12-15
12.7	Einsatz von Geräten und Terminals als Drucker.....	12-16
12.7.1	Geräte.....	12-16
12.7.2	Terminals.....	12-17
<b>A1</b>	<b>Anhang 1: I/O-Fehler-Codes.....</b>	<b>A1-1</b>
<b>A2</b>	<b>Anhang 2: Bandlaufwerk: Bedienung/Fehler-Codes.....</b>	<b>A2-1</b>
<b>A3</b>	<b>Anhang 3: Administrator-Kommandos.....</b>	<b>A3-1</b>
<b>A4</b>	<b>Anhang 4: Gerätedateien.....</b>	<b>A4-1</b>
<b>A5</b>	<b>Anhang 5: Dateiformate.....</b>	<b>A5-1</b>
<b>A6</b>	<b>Anhang 6: Stichwortverzeichnis.....</b>	<b>A6-1</b>

---



---

## Einleitung

---

### 1 Einleitung

Dieses Handbuch enthält Installations- und Bedienungsinformationen für das System TARGON /35. Unter dem Betriebssystem der TARGON /35 stehen Ihnen folgende UNIX-Versionen zur Verfügung:

- UNIX™ System V (UNIX ist ein eingetragenes Warenzeichen der Bell Laboratories)
- 4.2 BSD der University of California, Berkeley

Das Handbuch ist in folgende Sektionen aufgeteilt:

1. Kurzbeschreibung Hardware TARGON /35, in der die Hardware der Zentraleinheit und die anschließbare Peripherie vorgestellt wird.
2. Beschreibung des Console Operating Systems (COS) und dessen Handhabung.
3. Installation des Betriebssystems, in der die Installation des Root- und /usr-Dateisystems beschrieben wird.
4. In dem Kapitel über das Zwei-Welten-Konzept der TARGON /35 wird das Arbeiten mit UNIX System V und den Berkeley Extensions beschrieben.
5. Das Kapitel Systemverwaltung enthält wichtige Punkte und Vorschläge zur Systemverwaltung.
6. Im Kapitel Spool-System finden Sie die Beschreibung des Drucker-Spoolers der TARGON /35.

In den verschiedenen Anhängen finden Sie Beschreibungen zur Handhabung des Magnetbandlaufwerks sowie Fehlermeldungen und ihre Bedeutung. Außerdem finden Sie vollständige – jeweils alphabetisch geordnete – Beschreibungen der für die Administration wichtigen Kommandos, Gerätedateien und Dateiformate, soweit sie UNIX System V und die Erweiterungen der Nixdorf Computer AG betreffen. In den Beschreibungen der Kommandos, Gerätedateien und Dateiformate finden Sie in der Fußzeile jeder Seite die Bezeichnung des Moduls, auf das sich die Beschreibung bezieht.

## Einleitung

---

### System-Installation

Die Installation der TARGON /35 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 /35

### 2 Kurzbeschreibung Hardware TARGON /35

In diesem Kapitel werden die HW-Komponenten für das System TARGON /35, die mit Release 2 freigegeben werden, mit ihren Aufgaben und Eigenschaften im System beschrieben.

#### 2.1 Generelle Übersicht

Die Grundausstattung der TARGON /35 besteht aus folgenden Komponenten:

- Systemschrank
- CPU incl. 32 KB Daten-Cache
- 4 MB Speicher
- 1 I/O-Prozessor zum Anschluß von je einem Disk- und Tape-Controller sowie einem Systemdrucker
- 1 Disk-Controller zum Anschluß von 4 Magnetplattenlaufwerken
- 1 Tape-Controller zum Anschluß von 4 Magnetbandgeräten
- 1 System-Support-Prozessor (SSP) mit einer Systemkonsole, 1 Utility- und Mikrocode Floppy-Disk-Laufwerk und 1 Anschluß für Akustik-Koppler
- 1 Intelligenter Terminal Prozessor (ITP) zum Anschluß von 16 Bildschirmarbeitsplätzen

---

## Kurzbeschreibung Hardware TARGON /35

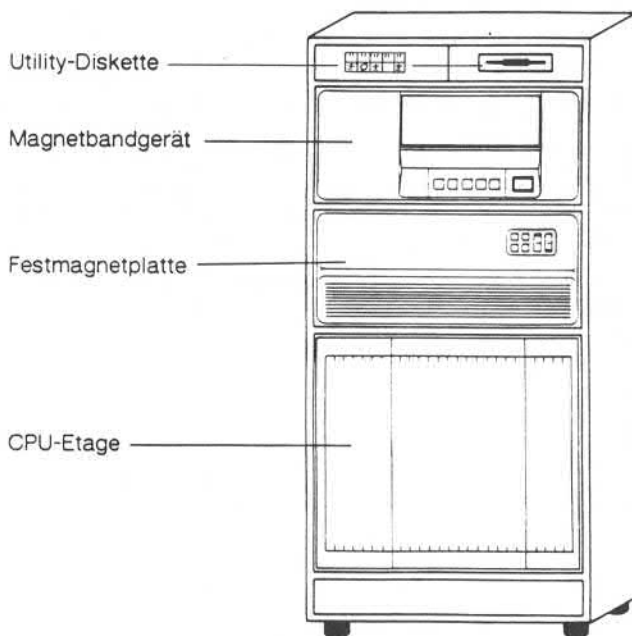
---

### 2.1.1 Systemschrank

Der Systemschrank der TARGON /35 beinhaltet in der Grundausstattung folgende Komponenten:

- CPU-Etage zur Aufnahme der kompletten CPU-Elektronik
- Disk-Etage mit einer Festmagnetplatte (Nettokapazität: 415 MB)
- Tape-Etage mit einem Magnetbandgerät 1600/3200 bpi, 100/50 ips
- die erforderlichen Netzteile für die max. benötigte Spannungsversorgung
- das Utility-/Mikrocode Floppy-Disk-Laufwerk
- die komplette Mechanik zur Aufnahme der Anschlußfelder

#### Systemschrank



## Kurzbeschreibung Hardware TARGON /35

### 2.1.2 Peripherieschrank

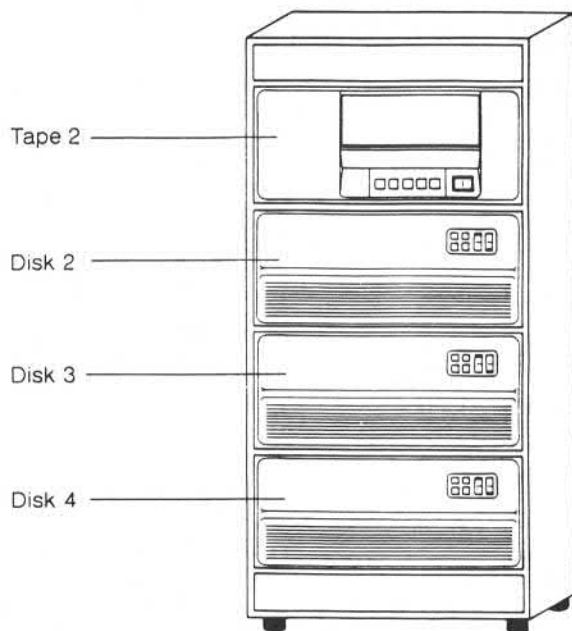
In der Grundausstattung der TARGON /35 sind bereits ein Magnetplatten- und ein Magnetbandlaufwerk im Systemschrank untergebracht.

Bei zusätzlicher Konfigurierung von Laufwerken werden Peripherieschränke nötig.

Bei der Erweiterung von Magnetplattenlaufwerken ist folgendes zu beachten:

- max. vier Laufwerke pro Disk-Controller,
- max. 8 Laufwerke je System.

#### Peripherieschrank



---

## Kurzbeschreibung Hardware TARGON /35

---

Für die Erweiterung der Magnetplatten- und Magnetbandausstattung werden zusätzliche Peripherieschränke benötigt.

Die größte Ausbaustufe der TARGON /35 ist ein Systemschrank und drei Peripherieschränke.

### 2.2 Prozessoren

Das System TARGON /35 verfügt über fünf funktionell selbstständige Prozessoren.

- Applikationsprozessor,
- Gleitkommaprozessor (optional),
- Intelligenter Terminalprozessor,
- Input/Output-Prozessor und
- System-Support-Prozessor.

## Kurzbeschreibung Hardware TARGON /35

### 2.2.1 Applikationsprozessor

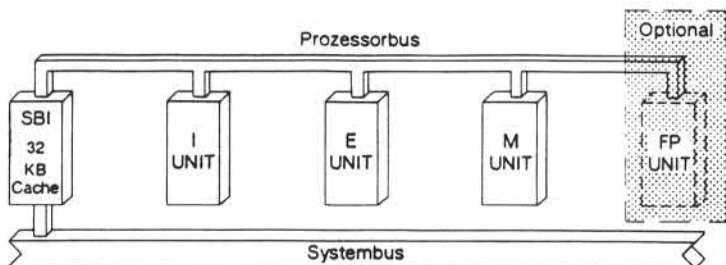
Der in Schottky-TTL-Logik ausgelegte Applikationsprozessor besteht aus vier Einschüben:

- Systembus-Interface (SBI) mit 32 KB Cache Speicher für Daten,
- Instruktionseinheit (Instruction-Unit oder I-Unit),
- Ausführungseinheit (Execution-Unit oder E-Unit),
- Mikrocode-Verwalter (Microcode-Sequencing-Unit oder M-Unit).

Die I-Unit holt Instruktionen aus dem schnellen 4 KByte großen Instruction-Cache und Operanden aus dem Register-Stack, Daten-Cache oder Speicher.

Die E-Unit führt die Instruktionen aus, wie sie als Mikroinstruktion vom Mikrocode-Verwalter definiert sind.

#### Prozessorausstattung



---

## Kurzbeschreibung Hardware TARGON /35

---

### 2.2.2 Gleitkommaprozessor

Der Applikationsprozessor kann auf Wunsch durch einen Gleitkommaprozessor zur Beschleunigung von Gleitkommaoperationen erweitert werden.

Diese Operationen sind auf der TARGON /35 sowohl für einfache als auch für doppelte Genauigkeit mit dem Standard-IEEE-Format möglich.

### 2.2.3 Intelligenter Terminal-Prozessor

Der Anschluß von lokalen oder remote Terminals erfolgt im System TARGON /35 über den Intelligenen Terminal-Prozessor (ITP). Jeder ITP verfügt über 16 asynchrone, bidirektionale RS-232C Kanäle. Für jeden Kanal können weiterhin mittels Wandlerkarte am Anschlußfeld alternativ RS-422- (V.11), 20mA- oder IHSS-Schnittstellen angeboten werden. Eine max. Point-to-Point-Leitungsverbindung (bei 9600 Bit/s) von 2000 m ist mit IHSS-Verbindungen möglich.

Die Leitungsgeschwindigkeit des ITP beträgt maximal 19.200 Bit/s pro Kanal. Der gesamte Durchsatz eines ITP beträgt 16.000 Zeichen/s.

Er besteht aus einer Arithmetik-Logik-Einheit, 4 K Worten beschreibbarem Kontrollspeicher, 4 KBytes festem RAM-Pufferspeicher und einem Interface zum System-Bus. Die Arithmetik-Logik-Einheit ist in schnellem Schottky-TTL implementiert mit einer Zykluszeit von 100 ns. Der ITP ermöglicht gleichzeitige Übertragungen mit 9600 Baud Geschwindigkeit auf allen 16 Leitungen.

Derzeit beträgt die höchste Anzahl ITP's je System 6 ITP's. Damit können z. Z. bis zu 95 Terminals betrieben werden. (Ein Anschluß ist für die Konsole reserviert.)



**Kurzbeschreibung Hardware TARGON /35**

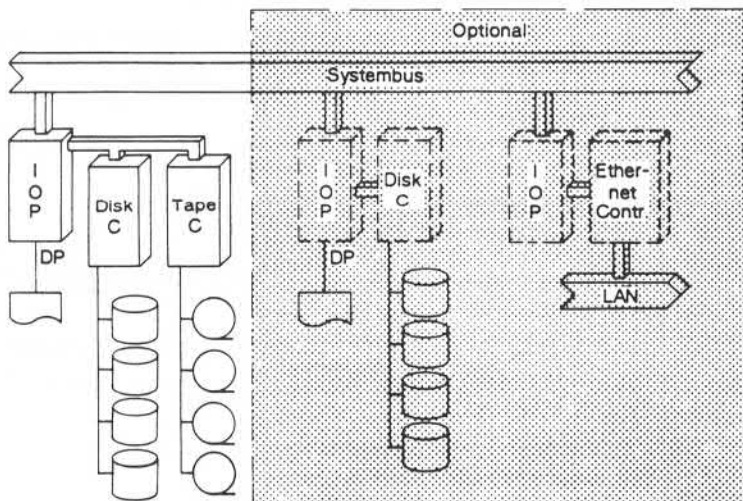
**2.2.4 Input/Output-Prozessor**

Der Input/Output-Prozessor (IOP) basiert auf einem 68000<sup>TM</sup> Mikroprozessor von Motorola. Die maximale Übertragungsrate je IOP beträgt 2,1 MB. Jeder IOP steuert SW-abhängig max. bis zu zwei Controller. Der IOP bietet darüber hinaus die Data Products Parallel-Schnittstelle zum Anschluß eines Systemdruckers.

Derzeit ist eine Konfigurierung von max. drei IOP's vorgesehen. Der erste IOP ist für die Steuerung der Disk- und Tape-Controller vorgesehen. Der zweite und dritte IOP wird zum einen benötigt, wenn mehr als vier Magnetplattenstationen (je Disk-Controller können bis zu vier Magnetplatten angeschlossen werden) konfiguriert werden und zum anderen, wenn ein Ethernet-Anschluß realisiert wird.

Der Tape-Controller ist dem ersten IOP zugeordnet und kann bis zu vier Streamer Magnetbandgeräte steuern. Der Anschluß erfolgt im Daisy-Chain-Verfahren.

**I/O-Prozessor**



© „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 Gebrauchsmusterantragung vorbehalten.“

---

## Kurzbeschreibung Hardware TARGON /35

---

### 2.2.5 System-Support-Prozessor

Der System-Support-Prozessor (SSP) ist ein intelligentes Subsystem. Die wesentlichen Komponenten sind:

1. Systemkonsole,
2. Utility-Diskette,
3. Akustik-Koppler.

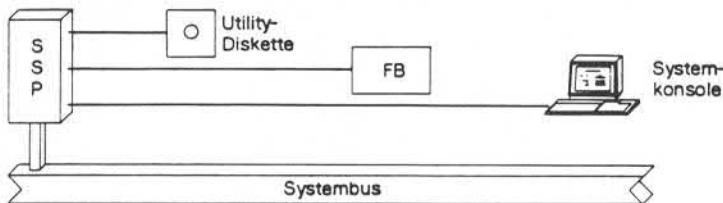
Während der Systeminitialisierung übernimmt der SSP folgende Aufgaben:

- Mikrocode der CPU und I/O-Prozessoren laden,
- Systemdiagnose,
- Testfunktion und
- Fehler-Report zur Systemkonsole.

Des weiteren können Systemkomponenten durch den Aufruf von Diagnose-Programmen im laufenden Betrieb getestet werden.

Die o. g. Funktionen werden durch den Fernbetreuungs-Anschluß und die 1,6 MB Utility-Diskette wesentlich unterstützt.

#### System-Support-Prozessor

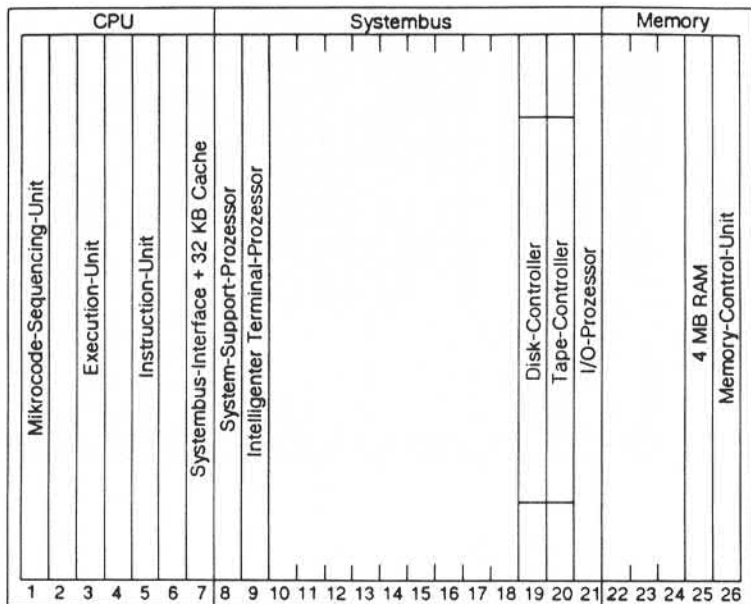


## Kurzbeschreibung Hardware TARGON /35

### 2.3 CPU-Etage

Die folgenden Abbildungen zeigen die Bestückung der CPU-Etage in der Grundausrüstung und in der Maximalausstattung:

#### CPU-Etage (Grundausrüstung)



---

 Kurzbeschreibung Hardware TARGON /35
 

---

## CPU-Etage (Maximalausstattung)

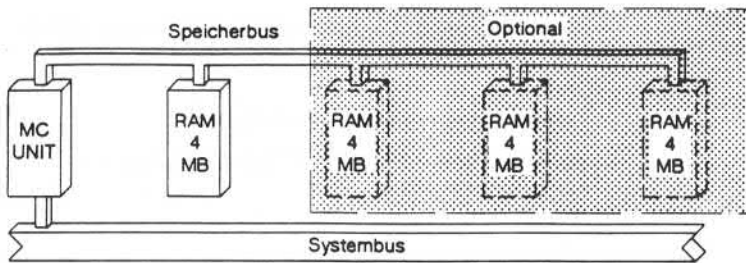
CPU							Systembus										Memory								
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
Mikrocode-Sequencing-Unit																									
Execution-Unit																									
Floating-Point-Unit																									
Instruction-Unit																									
Systembus-interface + 32 KB Cache																									
System-Support-Prozessor																									
Intelligenter Terminal-Prozessor																									
Intelligenter Terminal-Prozessor																									
Intelligenter Terminal-Prozessor																									
Intelligenter Terminal-Prozessor																									
Intelligenter Terminal-Prozessor																									
Intelligenter Terminal-Prozessor																									
Ethernet-Controller																									
I/O-Prozessor																									
Disk-Controller																									
I/O-Prozessor																									
Disk-Controller																									
Tape-Controller																									
I/O-Prozessor																									
4 MB RAM																									
4 MB RAM																									
4 MB RAM																									
4 MB RAM																									
Memory-Control-Unit																									

Kurzbeschreibung Hardware TARGON /35

2.4 Speicherausstattung

Die Speicherausstattung besteht aus einer Memory-Control-Unit, die den gesamten Speicherausbau verwaltet, und einem 4 MB Speichermodul. Ein fehlerkorrigierender Code bietet 1-Bit-Fehlerkorrektur und Mehr-Bit-Fehlererkennung.

Speicherausstattung



Die Speicherausstattung kann bei Bedarf durch 4 MB Einschübe erweitert werden. Die maximale Speicherkapazität beträgt 16 MB.

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

## Kurzbeschreibung Hardware TARGON /35

---

### 2.5 Peripherie

An das System TARGON /35 ist eine breite Palette von Peripheriegeräten anschließbar:

- Magnetplattenlaufwerke
    - Winchester Festplatten mit 475 MByte
  - Magnetbandgeräte
    - Cache-Streamer-Tape mit 1600/3200 bpi und 100/50 ips
  - Drucker
    - Zeilendrucker mit 600 Zeilen/Minute
    - serielle Drucker 75/150/210 Zeichen/Sekunde
  - Terminals
    - zeichenorientierte Terminals
    - Positivdarstellung
    - Terminalperipherie (Drucker)
  - Kommunikation
    - zeichenorientiert (V.11, V.24, IHSS)
-

---

## Console Operating System

---

### 3 Console Operating System (COS)

Das COS der TARGON /35 ist ein multifunktionales System, das unter dem System-Support-Prozessor (SSP) läuft. Dieses Kapitel beschreibt die Elemente des COS, die vom Systemadministrator genutzt werden.

Mit Hilfe von COS können Sie folgende Aufgaben ausführen:

- Laden des Mikrocodes der CPU und der E/A-Controller
- Kontrolle der CPU-Aktivitäten
- Booten der TARGON /35
- Kommunikation mit dem Betriebssystem
- Logging und Recovery von CPU-Fehlern
- Durchführung lokaler Diagnose und Ferndiagnose

Das COS wird von den Systemadministratoren zur Konfigurierung von Hardware, zur Installierung von Software und zum Booten und Überwachen des Betriebssystems benutzt. Die Spezialisten des Technischen Kundendienstes der Nixdorf Computer AG nutzen die Möglichkeiten von COS für Diagnosezwecke und Ausführung von Reparaturprogrammen.

#### 3.1 Diskettenlaufwerk

Zur Grundausstattung der TARGON /35 gehört das Utility- und Mikrocode-Floppy-Disk-Laufwerk. Die Diskette (1,6 MByte, 5¼") enthält das COS-Programm selbst sowie den Mikrocode der CPU und der E/A-Controller. Außerdem stehen separate Diagnosedisketten zur Verfügung. Beim Einschalten des Stroms kann die Diskette im Laufwerk sein, bevor Sie das System ausschalten, sollte die Diskette jedoch aus dem Laufwerk entfernt werden.

Der Inhalt der Diskette wird bei jedem Systemstart geladen. Dabei spielt es keine Rolle, ob der Strom ausgeschaltet war oder nicht. Läuft das Betriebssystem, stellt das COS die Möglichkeit zur Verfügung, Teile des Mikrocodes oder Diagnoseprogramme zu laden.

---

## Console Operating System

---

### 3.2 COS-Frames

Das COS erfüllt seine Funktionen mit Hilfe von 17 Schnittstellen-Programmen, den sogenannten Frames. Mit einer Ausnahme, dem Frame 0, geben die Frames ihre speziellen Funktionen aus. Frame 0 erscheint als „normaler“ Bildschirm und wird vom Systemadministrator als Schnittstelle zum Betriebssystem benutzt.

Die TARGON /35 hat zwei Betriebsarten:

- OPERATOR und
- CE (Customer Engineer).

Normalerweise arbeitet das System in der OPERATOR-Betriebsart. In dieser Betriebsart können Sie folgende Frames anwählen:

1. Index-Frame,
2. Frame 0 (Terminal-Mode) und
3. Frame 1 (Booting).

In dem Kapitel „COS-Frames für Systemadministratoren“ wird die Benutzung dieser Frames ausführlich erläutert.

In der CE-Betriebsart können Sie als Administrator auf folgende Frames zum Installieren des Betriebssystems und zur Konfigurierung von Platten zugreifen:

1. Frame 7 (E/A-Kanäle) und
2. Frame D (Hardware-Konfiguration).

Im Kapitel „Betriebssysteminstallation“ finden Sie nähere Erläuterungen zur Benutzung dieser Frames.

**Achtung:** Frames, die nur dem Technischen Kundendienst der Nixdorf Computer AG vorbehalten sind, sollten von Unbefugten nicht aufgerufen werden, da das eine erhebliche Beeinträchtigung des System-Laufzeitverhaltens zur Folge hat. In der Betriebsart CE kann der Versuch, diese Frames aufzurufen, zu einem Systemabsturz führen.



---

## Console Operating System

---

3

### 3.2.1 Systemstatus-Zeile

In allen Frames ist die unterste Zeile des Konsole-Bildschirms für Statusmeldungen des Systems und für den Frame-Status reserviert. Diese Zeile ist die Systemstatus-Zeile. Die Statusinformationen erscheinen in acht invertiert dargestellten Feldern. In jedem dieser acht Felder kann eine Nachricht ausgegeben werden, die bis zu acht Zeichen lang sein kann. Wird die Systemstatus-Zeile durch ein Programm zerstört, wird sie durch das COS wieder aufgebaut, sobald ein Frame angewählt wird. Im Kapitel „Nachrichten in der Systemstatus-Zeile“ finden Sie eine Liste der möglichen Meldungen sowie eine kurze Erläuterung zu jeder Meldung.

### 3.2.2 Auswahl der Frames

Sie können auf die Frames folgendermaßen zugreifen:

1. Aus jedem Frame können Sie durch Eingabe von <CTRL> und <^> in den Index-Frame wechseln. Dieser Index-Frame zeigt Ihnen am Bildschirm alle verfügbaren Frames mit ihrem Namen und ihrer alphanumerischen Identifikation an.
2. Durch Eingabe der alphanumerischen Identifikation können Sie in einen anderen Frame wechseln.

---

## Console Operating System

---

Layout des Index-Frames auf der Konsole

-Index-		Z-> Run/Stop
SELECT FRAME	(0)	COS TERMINAL MODE
	(1)	SYSTEM CONFIGURATION
	(2)	XTEND BUS INTERFACE TEST
	(3)	ALTER/DISPLAY STORAGES
	(4)	MICRO LOADER
	(5)	CPU TEST
	(6)	MACRO LOADER
	(7)	I/O CHANNEL TEST
	(8)	SYSTEM ENVIRONMENT
	(9)	MACRO DISASSEMBLER
	(A)	OPERATOR ENTERTAINMENT
	(B)	PROGRAM DEBUG
	(C)	ITP TEST
	(D)	HARDWARE CONFIGURATION
	(E)	MACHINE ERROR LOGOUT
	(F)	FUNCTION KEY PROGRAMMING
*-> CHANGE COS CHARACTER		
-CE MODE	CPU[0]	000FA020 COS" "1E *MANUAL*

Auf eine fehlerhafte Frameauswahl reagiert das COS durch ein akustisches Signal (Piepton) und gibt eine Fehlermeldung aus. Diese Fehlermeldungen finden Sie im Kapitel „Index-Frame“ näher beschrieben. Durch zweimalige Eingabe von <ESC> kehren Sie ebenfalls aus jedem anderen Frame – ausgenommen Frame 0 – in den Index-Frame zurück. Die Eingabe von <CTRL> und <^> ist die einzige Möglichkeit, Frame 0 zu verlassen.

---

## Console Operating System

---

### 3.2.3 Funktionsauswahl innerhalb eines Frames

In jedem Frame – ausgenommen Frame 0 – werden Ihnen die zur Verfügung stehenden Funktionen sowie das entsprechende Zeichen zur Anwahl dieser Funktion angezeigt. Nachdem Sie eine Funktion ausgewählt haben, wird diese Zeile blinkend dargestellt. Die Funktionszeilen haben folgendes Format:

X-> FUNKTION

wobei hier im Text das X das Zeichen symbolisiert, mit dem Sie die Funktion auswählen können, auf die der Pfeil zeigt.

Das COS reagiert auch hier auf eine fehlerhafte Funktionsauswahl durch ein akustisches Signal.

### 3.2.4 Cursor-Kontrolle und Dateneingabe

Die Arbeit mit einigen Frames erfordert – außer der reinen Funktionsauswahl – Cursorbewegungen und Dateneingaben. Haben Sie eine Funktion ausgewählt, befindet sich der Cursor automatisch im richtigen Feld. Durch Drücken der Taste <CR> bestätigen Sie die Eingaben, die Sie in den Feldern vornehmen. Haben Sie einmal fehlerhafte Daten eingegeben, die Sie noch nicht mit <CR> bestätigt haben, können Sie durch Betätigen der Taste <CAN> das Feld in den ursprünglichen Zustand zurückversetzen.

Das COS überprüft die Eingaben auf Richtigkeit (hexadezimal, dezimal, binär oder alphanumerisch) und meldet unkorrekte Eingaben durch ein akustisches Signal.

## Console Operating System

---

### 3.2.5 Der COS-Eingabepuffer

Das COS ist in der Lage, bis zu 32 Eingabezeichen – Neuanwahl von Frames, Operatoreingaben oder Daten – zu speichern. Sie können also bereits Eingaben tätigen, bevor die laufende Frame-Operation beendet ist. Der Eingabepuffer erlaubt Ihnen, Frame-Anwahlen und Operator-Eingaben, z. B. zum Restart des Betriebssystems in Frame 1, zu speichern.

### 3.3 Systemnachrichten

Zur Ausgabe von Nachrichten benötigt das Betriebssystem ein Terminal. Standardmäßig ist hierfür die Systemkonsole vorgesehen.

Nur wenn sich die Konsole im Frame 0 – dem Terminal-Modus – befindet, ist es dem Betriebssystem möglich, Nachrichten auszugeben. Befindet sich die Systemkonsole beim Senden von Nachrichten gerade in einem anderen Frame, werden diese Nachrichten solange gespeichert, bis sich die Konsole wieder im Frame 0 befindet. Ist dieser Nachrichtenspeicher zu 50 Prozent belegt, wird eine Warnung im zweiten Feld der Systemstatus-Zeile ausgegeben. Ist der Puffer voll, bevor Frame 0 wieder angewählt wurde, unterbricht das COS den aktuellen Frame (sofern es sich nicht um den Index-Frame handelt) und gibt die bis dahin gespeicherten Nachrichten aus.

Achtung: Die Konsole sollte sich **nie** für längere Zeit im Index-Frame befinden. Das Betriebssystem kann den Index-Frame nicht unterbrechen, um Nachrichten auszugeben.

---

---

## COS-Frames für Systemadministratoren

---

### 4 COS-Frames für Systemadministratoren

Dieses Kapitel beschreibt diejenigen Frames des COS, die Sie benötigen, um das System zu booten und mit dem Betriebssystem zu kommunizieren.

Diese Frames sind:

- Index-Frame,
- Frame 0 (Terminal-Mode) und
- Frame 1 (Booting).

In der Betriebsart CE können auch noch die Frames 7 (Installation des Betriebssystems) und D (Hardware-Konfigurierung) vom Systemadministrator benutzt werden. Alle anderen Frames sind für die Mitarbeiter des Technischen Kundendienstes der Nixdorf Computer AG reserviert.

#### 4.1 Index-Frame

Wie bereits in dem vorangegangenen Kapitel beschrieben, enthält der Index-Frame das Menü der vorhandenen Frames. Aus diesem Menü kann der gewünschte Frame ausgewählt werden.

Falsche Frameauswahl oder unkorrekte Operating-Eingaben werden durch ein akustisches Signal sowie durch die folgenden Meldungen angezeigt:

\*\*\*INVALID FRAME NUMBER\*\*\*

Das angegebene Frame-Identifikationszeichen ist ungültig.

\*\*\*MUST BE IN CE MODE\*\*\*

Der angegebene Frame kann nur angesteuert werden, wenn sich das System in der CE-Betriebsart befindet.

---

## COS-Frames für Systemadministratoren

---

\*\*\*FRAME REMOTELY N/A\*\*\*

Die Remote-Konsole ist durch die Operating-Eingabe X in Frame 1 nicht verfügbar. Aus diesem Grund kann der ausgewählte Frame nicht angesteuert werden.

\*\*\*FRAME CURRENTLY N/A\*\*\*

Der ausgewählte Frame ist zur Zeit nicht verfügbar.

### 4.1.1 Ändern des COS-Schlüssels

Das Frame-Identifizierungszeichen wird im Index-Frame als COS-Schlüssel (COS character key) signifiziert. Standardschlüssel sind die Tasten <CTRL> und <^>; kann aber durch jede andere Eingabe außer <\*>, <ESC> oder <CR> ersetzt werden. Um den aktuellen COS-Schlüssel zu ändern, müssen Sie in den Index-Frame (durch Drücken der Tasten <CTRL> und <^>) gehen. Sie sehen dann in der Ausgabe folgende Zeile:

\*-> CHANGE COS CHARACTER

Durch Eingabe von \* und des gewünschten Schlüssels können Sie nun einen anderen Frame anwählen. Ihre Eingabe erscheint als hexadezimaler Wert im dritten Feld der Systemstatus-Zeile. Haben Sie einen der nicht erlaubten Schlüssel eingegeben, ertönt ein akustisches Signal und folgende Meldung wird ausgegeben:

\*\*\*INVALID COS CHARACTER\*\*\*

## COS-Frames für Systemadministratoren

### 4.2 Frame 1

Frame 1 wird zum Booten, für den Autoboot und für System-Restarts genutzt.

Bildschirmausgabe Frame 1 – Systemkonfiguration

<<ESC F#/OPCTL>>		-SYSTEM CONFIGURATION-	Z-> RUN/STOP
<b>FEATURES</b> FLAG=00000000 SSL STATIC AUTOBOOT ON CPU(s) [0] COS R 9.0 MCODE F0 FP ACCEL 32K CACHE IOC OC 0111D3	<b>STORAGES</b> 8M MAIN 64K CONTROL 2K STK CACHE 1M SSP FDU	<b>AVAILABLE I / O</b>  CHANNELS Intelligent Terminal Proc(40) Standard Channel (0C)  <b>PERIPHERALS</b>  CH(s)DV-----NAME----- OC(0)00 Eagle 410D	<b>OPERATOR CONTROLS</b> A-> SET AUTOBOOT S-> SET TARGSTEP R-> SYS RESET B-> BOOT P-> START PID 0000 I-> INITIAL RST D-> BOOT DEVICE PER [0C(00)00] +-> IMPL MODULES M-> MODEM BAUD 1200 U-> SET BOOT FLG T-> SSL OPTION C-> RST CHANNELS X-> DISABLE REMOTE Q-> COLOR SCHEME V-> VIRTUAL CONSOLE CNTRL CPU [0] L-> PRINTER OFF
-CE MODE	CPU[0]	000FA020	COS" "1E
			*MANUAL*

Die Daten der Konfiguration – außer Q und V – sind auf der Diskette gespeichert. Um das System entsprechend der Daten in Frame 1 zu re-konfigurieren, müssen Sie das System aus- und wieder einschalten.

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



---

## COS-Frames für Systemadministratoren

---

### 4.2.1 Frame-Auswahl und CPU-Kontrolle in Frame 1

Die Kopfzeile in Frame 1 unterstützt Funktionen zur Frame-Auswahl und zur CPU-Kontrolle:

ESC F#/OPCTL Diese Komponente unterstützt die Frame-Auswahl. Durch Eingabe von <ESC> 0 gelangen Sie – unter Umgehung des Index-Frames – in Frame 0. /OPCTL erlaubt Ihnen die meisten Operatoreingaben, ohne erst Frame 1 anzuwählen. Z. B. können Sie in Frame 1 den Bootvorgang durch die Eingabe von B oder b starten, aus jedem anderen Frame geschieht dies durch die Eingabe von <ESC> 1/b. Auf diese Weise können Sie auf die Funktionen B, V, S, R, I und C aus jedem Frame zugreifen. Eine Fehleingabe wird durch Einblendung des Index-Frames quittiert.

[X] Z-> RUN/STOP Diese Komponente startet und stoppt die CPU. X ist die CPU-Nummer. In einer Single-Prozessorumgebung ist X=0; in einer Mehrprozessor-Umgebung ist X die virtuelle Prozessornummer, die durch die Operatoreingabe V vom Administrator festgelegt wird. Haben Sie durch die Operatoreingabe S die CPU in den RUN/STOP-Modus gesetzt, ändert die Eingabe von Z den aktuellen CPU-Status. Den aktuellen Status sehen Sie im letzten Feld der Systemstatus-Zeile. Der Status wird entweder als \*MANUAL\* oder RUNNING ausgegeben. Sind Sie durch Eingabe von S im INST STEP Modus, erlaubt Ihnen die Eingabe von Z die Ausführung eines Maschinenbefehls.



## COS-Frames für Systemadministratoren

### 4.2.2 Features in Frame 1

Die erste Spalte in Frame 1 – mit FEATURES überschrieben – unterstützt die folgenden Funktionen:

Feature	Funktion
FLAG=000000XX	000000XX ist die Boot-Option, die init beim Booten übergeben wird. Die Eingabe dieser Option kann nach der Eingabe von U erfolgen; der Cursor steht dann im richtigen Feld.

Folgende Boot-Optionen stehen zur Verfügung:

00000000 – Die Anlage wird automatisch in den Multi-User-Betrieb hochgefahren. Vorher erfolgt die Überprüfung der Dateisysteme.

00000001 – Während des Bootvorgangs wird die interaktive Eingabe der Adresse des Boot-Geräts sowie des Kernelnamens verlangt. Das System wird in den Multi-User-Betrieb hochgefahren.

00000002 – Das System wird in den Single-User-Betrieb hochgefahren (Standardoption).

00000003 – Kombination der Optionen 1 und 2. Sie müssen Boot-Gerät und Kernelnamen interaktiv eingeben; das System fährt dann hoch bis in den Single-User-Betrieb.

00000020 – Im Notfall oder beim Rebooten wird der Kernel-Debugger aufgerufen. Dies setzt voraus, das der Kernel mit Debugger-Option ausgestattet ist.

---

**COS-Frames für Systemadministratoren**

---

<b>Feature</b>	<b>Funktion</b>
SSL STATIC/ROTATE	Dieser Schalter wird durch die Eingabe von T betätigt. Setzen Sie den Schalter auf STATIC, wird die Systemstatus-Zeile der CPU unter Konsole-Kontrolle angezeigt (Operatoreingabe V). Das Setzen des Schalters auf ROTATE ändert die Ausgabe der Systemstatus-Zeile für eine Multi-Prozessorumgebung. Alle zwei Sekunden wird die Systemstatus-Zeile des nächsten Prozessors angezeigt. Im dritten Feld der Zeile steht die entsprechende Prozessornummer.
AUTOBOOT ON/OFF	Autoboot-Status ist ein oder aus. Diesen Schalter betätigen Sie durch Eingabe von A.
CPU(s) [X]	In einer Single-Prozessorumgebung ist X = 0. In einer Multi-Prozessorumgebung bezeichnet X die Prozessornummer. Mit der Operatoreingabe V können Sie diese Nummer spezifizieren.
COS R XX	XX ist die Bezeichnung des COS-Release.
MODE XXX	XXX ist die Bezeichnung des Mikrocode-Release.
FP ACCEL	Das System unterstützt hardwaremäßig die Beschleunigung der Floating Point Unit.
32K CACHE	Das System ist mit einem 32 KB großen Cache-Speicher ausgestattet.
IOC 0X XXXXXX	XXXXXX ist die hexadezimale Ausgabe des IOC-Mikrocode-Release. 0X ist die IOC-Nummer. Für jeden konfigurierten IOC erscheint solch ein Eintrag.

---

## COS-Frames für Systemadministratoren

---

### 4.2.3 Anzeige der Speicher in Frame 1

In der Spalte, die mit STORAGES überschrieben ist, finden Sie Informationen über die Größe des Hauptspeichers, Speicher für Mikromaschine und Zielmaschine und die Speicherklassen und -größen des System-Support-Prozessors.

### 4.2.4 Anzeige der I/O-Geräte in Frame 1

Unter der Rubrik CHANNELS finden Sie Nummern und Typen der konfigurierten IOCs und ITPs.

Unter der Überschrift PERIPHERALS finden Sie die Kanalnummer, die Subkanalnummer sowie Namen und Nummern der konfigurierten Plattenlaufwerke.

---

## COS-Frames für Systemadministratoren

---

### 4.2.5 Operatoreingaben in Frame 1

Unter der Überschrift OPERATOR CONTROLS finden Sie die gültigen Eingaben aufgelistet:

Eingabe	Funktion
A-> SELECT AUTOBOOT	Mit diesem Schalter schalten Sie den Autoboot ein oder aus. Die aktuelle Schalterstellung wird jeweils auf die Diskette geschrieben. Steht der Schalter beim Einschalten des Systems oder bei der Anwahl von SYS RESET in Frame 1 auf ON, startet das COS automatisch aus Frame 1 das Betriebssystem. Die Auswahl der Autoboot-Optionen in Frame 1 legt den Ablauf des vom COS initiierten Bootvorgangs fest.
S-> SET TARGET STEP	Diese Eingabe setzt den CPU-Modus auf RUN/STOP oder INST STEP. Die Eingabe von Z versetzt die Anlage in den Zustand wie im Kapitel „Frame-Auswahl und CPU-Kontrolle“ beschrieben.
R-> SYS RESET	Reset der Zielanlage.
B-> BOOT	Diese Eingabe ruft I, C und R auf. Anschließend wird das Bootstrap-Programm vom Boot-Gerät geladen. Die Anlage verbleibt anschließend im Status MANUAL.  Im Feld 7 der Systemstatus-Zeile werden Nachrichten über den Bootvorgang ausgegeben:  +BOOTING wird während des Ladevorgangs des Bootstrap-Programms angezeigt.  <BOOTED> erscheint nach dem erfolgreichen Laden des Bootstrap-Programms.

**COS-Frames für Systemadministratoren**

Eingabe	Funktion
	<p>*NO BOOT signalisiert einen Fehler während des Ladens.</p> <p>Nach dem Ladevorgang wird das Bootstrap-Programm durch Eingabe von Z ausgeführt.</p>
P-> START PID	Diese Eingabe startet den Prozeß mit der angegebenen Nummer. Standardwert ist 0000.
I-> INITIAL RESET	Reset der Mikromaschine. Leeren des Hauptspeichers und Anlegen des ersten MByte vom virtuellen Speicher.
D-> BOOT DEVICE	Unterstützt die Eingabe der Adresse des Geräts, von dem das COS das Bootstrap-Programm laden soll. Nach Eingabe von D befindet sich der Cursor in dem Feld, in dem die Adreßeingabe erfolgen muß. Jetzt können Sie die Geräteadresse eingeben; die Eingabe muß mit <CR> bestätigt werden. Wurde die Eingabe noch nicht bestätigt, können Sie sie durch Drücken der Taste <CAN> wieder rückgängig machen. Die Adresse besteht aus drei Feldern, der IOC Systembusadresse (Standard = 0x0c), der Subkanalnummer (Platte = 0, Band = 1) und der Nummer des Laufwerks (Standard = 0).

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

---

**COS-Frames für Systemadministratoren**


---

<b>Eingabe</b>	<b>Funktion</b>
*-> IMPL MODULES	<p>Diese Funktion stellt Ihnen acht Optionen zur Verfügung:</p> <p>0-&gt; IMPL SYSTEM ONLY</p> <p>1-&gt; IMPL CONTROL STORE ONLY</p> <p>2-&gt; IMPL PICO STORE ONLY</p> <p>3-&gt; IMPL LOCAL STORE ONLY</p> <p>4-&gt; IMPL IOCs ONLY</p> <p>5-&gt; IMPL GPIOs ONLY</p> <p>6-&gt; IMPL MCIOs ONLY</p> <p>7-&gt; ORIGINAL SYSTEM RESTART</p> <p>Anmerkung: Control Store und Pico Store sind CPU-Speicher, die mit Mikrocode geladen werden. Bei dem Pico Store handelt es sich um einen kleineren Speicher, der nur Mikrocode für die E-Unit enthält.</p>
U-> SET BOOT FLG	<p>Diese Eingabe ermöglicht Ihnen das Setzen der Boot-Optionen in der Spalte FEATURES. Nach Eingabe von U steht der Cursor automatisch im richtigen Feld für die Eingabe der Boot-Option.</p>
T-> SSL OPTION	<p>Durch diese Eingabe schalten Sie die Systemstatus-Zeile zwischen STATIC und ROTATE hin und her.</p>
C-> RESET CHANNELS	<p>Durch Eingabe von C wird nach einem Reset des Systembusses ein Reset der Kanäle vorgenommen.</p>
X-> ENABLE/DISABLE REMOTE	<p>Durch Eingabe von X wird die Remotefähigkeit des COS gesetzt oder gesperrt.</p>

## COS-Frames für Systemadministratoren

Eingabe	Funktion
V-> VIRT. CONSOLE	Diese Eingabe wird nur in einer Multi-Prozessorumgebung benötigt. Sie können hierdurch – mit der entsprechenden Nummer – den Prozessor benennen, der unter Konsol-Kontrolle laufen soll. Der CPU-Eintrag in der Spalte FEATURES zeigt Ihnen die Nummern der installierten Prozessoren. Nochmaliges Eingeben von V aktiviert die unter den Operator-Eingaben liegende Liste; die Nummern 0 – 3 werden ausgegeben. In einer Single-Prozessorumgebung sollte die Nummer auf 0 gesetzt sein. Jede andere Nummer bewirkt, daß der Prozessor nicht auf die Konsol-Kontrolle reagiert.



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

### 4.3 Frame 0

Frame 0 ist der Frame für den Terminal-Modus. Er fungiert als Schnittstelle zwischen Konsole und Betriebssystem und sollte eingeschaltet werden, sobald die Anlage gebootet ist und arbeitet, da die Ausgabe von Meldungen des Betriebssystems nur in Frame 0 erfolgt.

#### 4.3.1 Terminalstatus-Zeile

In Frame 0 ist die erste Zeile des Bildschirms als Status-Zeile reserviert. Sie wird unterstrichen dargestellt und enthält Informationen über das Terminal, z. B. ob es im Duplex-Betrieb arbeitet oder ob nur Großschreibung möglich ist. Folgende Informationen werden ausgegeben:

FDX    Terminal arbeitet im Duplex-Betrieb.

HDX    Terminal arbeitet im Halb-Duplex-Betrieb.

## COS-Frames für Systemadministratoren

---

### 4.3.2 Verlassen des Frame 0

Durch Drücken der Tasten <CTRL> und <↵> können Sie Frame 0 verlassen. Sie befinden sich dann wieder im Index-Frame. Sie erinnern sich, daß Sie in anderen Frames durch Eingabe von <ESC> und der neuen Frame-Nummer den aktuellen Frame verlassen und ohne Umweg über den Index-Frame in einen anderen Frame wechseln können. In Frame 0 haben Sie diese Möglichkeit nicht.

### 4.3.3 Abbruch von Frame 0

Die Eingabe von <CTRL> und <Local> bewirkt die sofortige Rückkehr in den Index-Frame. Sie sollten diesen abrupten Wechsel nur benutzen, wenn keine andere Möglichkeit besteht, Frame 0 zu verlassen.

### 4.3.4 Verwendung des COS-Schlüssels aus Programmen

Falls ein Benutzerprogramm den COS-Schlüssel als Eingabe benutzt, kann das COS das nicht unterscheiden und interpretiert die Eingabe als Aufforderung zum Wechsel in den Index-Frame. In diesem Fall kann der Operator den COS-Schlüssel im Index-Frame ändern. Anschließend wird wieder in Frame 0 gewechselt. Die Ausgaben, die vorher auf der Konsole waren sind dann allerdings verloren.

---



## Nachrichten in der Systemstatus-Zeile

### 5 Nachrichten in der Systemstatus-Zeile

Dieses Kapitel beschreibt die Meldungen, die das COS in den acht Feldern der Systemstatus-Zeile ausgibt. Diese Nachrichten geben Ihnen Informationen über die Aktivitäten des COS. Die möglichen Meldungen in jedem Feld finden Sie in der nachfolgenden Tabelle. Meldungen, die hier *kursiv* gedruckt sind, symbolisieren lediglich den Typ einer Meldung. Die jeweils angezeigte Nachricht kann also von den unten aufgeführten Überbegriffen abweichen.

Feld	Nachrichten
1	-CE MODE, *REMOTE*
2	<i>BUSERRS</i> , BUF 50 %, BUF FULL, BUS HUNG, <i>CTR</i> , N/ACC#XX, <i>SSPERRS</i>
3	CPU BUSY, CPU [X]
4	<i>FPYST</i> , <i>MICRO</i>
5	<i>FPYADR</i> , <i>MACRO</i>
6	<i>COS"/'XX</i>
7	BAD DISK, BAD UIB?, <BOOTED>, +BOOTING, <i>IOCERRS</i> , <i>IOCXXNR</i> , *NO BOOT, NO IMPL, NO RESET, PGFLT0-2
8	***CHECKSTOP*** (wird horizontal gescrollt), STOPPED, COS READY, -HALTED-, -HRDSTOP, *MANUAL*, MCK REC, NO MANUL, +RUNNING

Die Felder 1, 2, 4, 5 und 7 können leer sein. Die Felder 3, 6 und 8 enthalten immer eine Meldung. Eine Erläuterung zu jeder Meldung finden Sie in den folgenden Kapiteln.



---

## Nachrichten in der Systemstatus-Zeile

---

### 5.1 Meldungen in Feld 1

Die beiden Meldungen, die in Feld 1 ausgegeben werden, haben folgende Bedeutung:

- |          |  |
|----------|--|
| -CE MODE | Schlüsselstellung am Frontpanel des Systemschranks ist C.E.ON (CUSTOMER ENGINEER-Modus). In dieser Betriebsart ist es den Mitarbeitern des Technischen Kundendienstes der Nixdorf Computer AG möglich, auf alle Frames zuzugreifen, um alle notwendigen Arbeiten durchzuführen. Im Normalbetrieb sollten Sie in dieser Betriebsart nicht arbeiten. |
| +REMOTE  | Die Remote-Konsole ist aktiviert. Alle Kommandos sind im CE-Modus.   |

### 5.2 Meldungen in Feld 2

Die Nachrichten in Feld 2 haben die folgende Bedeutung:

- |                |  |
|----------------|--|
| BUF 50 %       | Der Puffer für Systemnachrichten ist bereits zu 50 % gefüllt. Der Puffer wird immer dann in Anspruch genommen, wenn sich das COS nicht in Frame 0 befindet. Sie erinnern sich, daß Systemnachrichten <b>nur</b> in Frame 0 ausgegeben werden. Ist das COS in einem anderen Frame, werden die anfallenden Meldungen in dem Puffer zwischengespeichert, bis Frame 0 ausgewählt wird. |
| BUF FULL       | Aus dem aktuellen Frame wird automatisch in Frame 0 gewechselt und der Inhalt des vollen Speichers wird am Bildschirm ausgegeben.  |
| BUS HUNG       | Der System-Support-Prozessor versucht, Codes an den Systembus zu senden. Dieser ist jedoch nicht empfangsbereit.   |
| <i>BUSERRS</i> | Hexadezimale Anzeige von Systembus-Fehlern.  |

---

### Nachrichten in der Systemstatus-Zeile

---

<i>CTR</i>	Das COS versucht, die CPU für eine Tastatureingabe zu stoppen. Die Anzahl der Versuche wird im Wiederholungszähler gespeichert. Der Inhalt dieses Zählers wird angezeigt.
<i>N/ACC#XX</i>	Der Systembus kann auf das Gerät XX nicht zugreifen. XX kann folgende Werte annehmen:  00 CPU 0C I/O-Controller 0D I/O-Controller 0E I/O-Controller  40-45 ITPs
<i>SSPERRS</i>	Ausgabe von SSP-Fehlerarten.

### 5.3

#### Meldungen in Feld 3

Im Feld 3 kann eine der beiden folgenden Meldungen erscheinen:

CPU BUSY	Diese Nachricht erscheint, wenn der Wiederholungszähler (s. auch Meldung <i>CTR</i> in Feld 2) 1024 Versuche gezählt hat, die CPU zu stoppen.
CPU [X]	Zeigt die Nummer der CPU an, deren Status gerade in der Systemstatus-Zeile dokumentiert wird.



---

## Nachrichten in der Systemstatus-Zeile

---

### 5.4 Meldungen in Feld 4

Die in Feld 4 ausgegebenen Nachrichten haben folgende Bedeutung:

*FPYST* Hier wird der Status des Diskettenlaufwerks angezeigt: reading, writing, seeking, erkannte Schreib- oder Lesefehler und NO DISK. Die Nachrichten werden nur während eines Zugriffs auf die Diskette ausgegeben.

*MICRO* Status des Programmzählers für die Mikromaschine.

### 5.5 Meldungen in Feld 5

Folgende Nachrichten erhalten Sie in Feld 5:

*FPYADR* Hier wird Ihnen Spur und Sektor der Diskette ausgegeben. Die Anzeige erfolgt in der Form TsXXSCYY:

s Seite a oder b der Diskette

XX Spur 00-4C

YY Sektor 00-1A

Diese Nachrichten erscheinen nur während eines Zugriffs auf die Diskette.

*MACRO* Status des Programmzählers der Makromaschine. Diese Nachricht erscheint nur, wenn der Mikrocode gestoppt ist.

## Nachrichten in der Systemstatus-Zeile

### 5.6 Meldungen in Feld 6

In Feld 6 finden Sie die folgenden Informationen:

*COS"C"XX* "C"XX symbolisiert den COS Schlüssel in folgender Form:

*C* = aktueller COS Schlüssel

*XX* = Hexadezimaler Wert des ASCII-Zeichens

Z. B. ist die Anzeige für den Standardschlüssel (<CTRL> und <^>) *COS"|1E*, da 1E der hexadezimale Wert dieser Tasten ist. Die Anzeige *COS"0"30* bedeutet, daß der aktuelle Schlüssel 0 mit dem hexadezimalen Wert 30 ist.

### 5.7 Meldungen in Feld 7

Folgende Informationen erscheinen in Feld 7:

*BAD DISK* Beim Laden der Diskette wurde ein Fehler im Mikrocode festgestellt.

*BAD UIB?* Beim Senden eines Unit Initialization Blocks (UIB) antwortet der Kanal entweder gar nicht oder er sendet eine Nachricht, daß der UIB fehlerhaft ist. Der UIB ist eine Tabelle auf der Diskette, die die Physik der angeschlossenen Plattenlaufwerke beschreibt. Für jedes konfigurierte Laufwerke muß ein UIB definiert sein.

*<BOOTED>* Der IOC hat den Bootblock in den Hauptspeicher gelesen. Das System kann nun durch Drücken der Taste Z gestartet werden. Diese Meldung erscheint in Frame 1.

*+BOOTING* Der Bootblock wird in den Hauptspeicher geladen. Diese Meldung erscheint in Frame 1.

*IOCERRS* Von einem I/O-Kanal zurückgelieferter Fehlercode.

*IOCXXNR* Der I/O-Kanal XX ist im Status NOT READY.



---

### Nachrichten in der Systemstatus-Zeile

---

*NO BOOT	Der I/O-Controller kann den Bootblock nicht vollständig in den Hauptspeicher einlesen.
NO IMPL	Der SSP hat von einer Einheit die Meldung „no acceptance“ erhalten. Die Einheit(en), für die das IMPL (Initial Micro Program Loading) fehlgeschlagen ist, ist nicht bekannt. Sie sollten einen Restart des Systems vornehmen.
NO RESET	Der SSP hat versucht, den Mikrocode der CPU neu zu setzen. Dieser Versuch ist mißlungen. Sie können davon ausgehen, daß entweder die CPU nicht läuft oder der Mikrocode fehlerhaft ist.
PGFLT0-2	Seitenfehler in Level 0, 1 oder 2 bei der Übersetzung virtueller Adressen in reale Adressen.

### 5.8 Meldungen in Feld 8

In Feld 8 werden die folgenden Informationen ausgegeben:

**\*\*\*CHECKSTOP\*\*\***

Die Überprüfung der Anlage hat stattgefunden, trotzdem konnte ein Restart der Anlage nicht vorgenommen werden.

**STOPPED** Die CPU hat einen rekursiven Status erkannt. Aufgrund dieses Status stoppt die CPU.

**COS READY** Diese Nachricht erscheint nur nach einem Restart der Anlage oder nach einem Reset des COS durch den Reset-Knopf am Frontpanel. Die Meldung zeigt an, daß der Eingabespeicher für die Datenaufnahme bereit ist.

**-HALTED-** Die Anlage hat ein HALT-Kommando empfangen. Die Anlage kann nicht einfach durch Eingabe von Z wieder gestartet werden.

**-HRDSTOP** Die Maschinenuhren werden gestoppt und der Emulator befindet sich in der Mitte einer Zielinstruktion.

### Nachrichten in der Systemstatus-Zeile

*MANUAL*	Die Emulation der Zielinstruktion ist beendet. Die Anlage ist vor dem Beginn der Emulation der nächsten Instruktion.
NO MANUAL	Die CPU hat auf eine STOP-Aufforderung des COS nicht reagiert. Timeout ist eine Sekunde.
MCK REC	Die Anlage wird überprüft. Die Ergebnisse werden auf der Diskette aufgezeichnet.
+RUNNING	Die Anlage läuft.

### 5.9 Ausgabe der Systemaktivitäten

An der Vorderseite des Systemschranks befindet sich eine Leuchtanzeige, die aus zehn LEDs besteht. Diese Anzeige spiegelt einige der Systemaktivitäten der TARGON /35 wider, die in dem Kapitel „Meldungen in Feld 8“ beschrieben sind. Die Anzeige wird durch das COS gesteuert und aktualisiert. Ist das COS voll ausgelastet, fehlen die Kapazitäten, um das Leuchtfeld zu aktualisieren und die Anzeige erlischt zeitweise.

Die Leuchtanzeige hat folgende Bedeutungen:

**\*\*\*CHECKSTOP\*\*\***

Die einzelnen Teile der Leuchtanzeige leuchten fortlaufend. Es beginnt mit den mittleren beiden LED's und setzt sich zur rechten und linken Seite fort. Wenn alle LED'S aufleuchten, erlischt die Anzeige und das Muster wird aufs Neue aufgebaut.

**STOPPED**

Das LED an der rechten Seite der Leuchtanzeige beginnt zu leuchten. Das Leuchten wandert über alle anderen LED's bis zur linken Ecke und beginnt dann wieder an der rechten Seite (horizontales Scrolling).

**HALTED**

Die LED's leuchten nacheinander – beginnend in der rechten Ecke – auf. Wenn alle LED's leuchten, werden sie nacheinander – wieder beginnend in der rechten Ecke – gelöscht, und das Muster wird wieder aufgebaut.



---

### Nachrichten in der Systemstatus-Zeile

---

HRDSTOP	Die bestehende Leuchtanzeige bleibt erhalten.
MANUAL	Zwei LED's beginnen in der rechten und linken Ecke der Anzeige zu leuchten. Das Leuchten wandert bis zur Mitte der Anzeige, dann beginnt das Scrollen von den Ecken bis zur Mitte wieder.
+RUNNING	Die Leuchtanzeige repräsentiert die Auslastung des Systems durch Benutzer und Kernel. Von den zehn Leuchtdioden der Anzeige sind fünf erleuchtet. Jedes der leuchtenden LED's symbolisiert 20 % der Systemauslastung. Die linke Hälfte der Anzeige ist für die Kernelauslastung reserviert, die rechte für die Benutzer- auslastung.

---



## COS-Reset und System-Restart

### 6 COS-Reset und System-Restart

Dieses Kapitel beschreibt die Vorgänge bei einem COS-Reset und einem System-Restart.

Ein COS-Reset können Sie vornehmen, in dem Sie den RESET-Schalter am Frontpanel betätigen. Ein System-Restart wird nach dem Stromeinschalten vorgenommen. Sie können einen System-Restart aber auch durchführen, indem Sie in Frame 1 den \* (IMPL MODULES) gefolgt von einer 7 (SYSTEM RESTART) eingeben.

#### 6.1 Abläufe beim COS-Reset

Der RESET-Schalter am Frontpanel wird benutzt, um das COS mit der Zielmaschine und den Mikromaschinen zu synchronisieren. Durch das Betätigen des RESET-Schalters werden folgende Vorgänge gestartet:

1. Der Inhalt des SSP-Speichers wird gelöscht.
2. Das COS und die unter COS laufenden Programme werden von der Diskette geladen.
3. Frame 1 wird geladen. Die Kontrolle wird Frame 1 übergeben, der sich jetzt im Wartezustand befindet. Das COS aktualisiert die Systemstatus-Zeile und gibt die Meldung COSREADY aus.

---

## COS-Reset und System-Restart

---

### 6.2 Abläufe beim System-Restart

Beim Restart des Systems aus Frame 1 wird keine Überprüfung der Systemkomponenten vorgenommen. Wird der Restart des Systems durch Stromeinschalten vorgenommen, wird diese Überprüfung durchgeführt. In den Feldern 4 und 5 der Systemstatus-Zeile wird angezeigt, welche Tests gerade ablaufen. Nach jedem erfolgreichen Abschluß eines Tests wird in der Mitte des Bildschirms ein Stern (\*) ausgegeben. Die Sterne werden horizontal von rechts nach links aneinandergereiht. Neun Sterne nebeneinander signalisieren die erfolgreiche Beendigung aller Tests.

Der Testdurchlauf wird auch auf der Leuchtanzeige in der Fronttür des Systemschranks ausgegeben. Beim Testbeginn leuchten alle Lampen auf. Sie werden dann sukzessive nach Beendigung der Tests von rechts nach links gelöscht.

1. Das COS-Bootprogramm führt diverse Initialisierungsprogramme aus:
  - Die Diagnostetests werden durchgeführt. Folgende Komponenten werden überprüft: Leuchtanzeige für die Systemaktivitäten, Konsole-, Drucker-, Modem- und Systembuskommunikation, Speicheradressen und -zellen sowie Diskettenzugriff.
  - Verläuft einer der Tests negativ, muß das System erneut aus- und wieder eingeschaltet werden.
  - Der Inhalt des SSP-Speichers wird gelöscht.
  - Das COS und die unter COS laufenden Programme werden geladen.
  - Frame 1 wird geladen und die Kontrolle übergeben.

---

## COS-Reset und System-Restart

---

2. Frame 1 lädt das Initialisierungs-Mikroprogramm (IMPL). Es wird eine Nachricht ausgegeben, daß der Mikrocode geladen wird; außerdem werden Sie durch weitere Meldungen über den Fortschritt des Ladens unterrichtet. Folgende Arbeiten werden durchgeführt:
    - Laden des Kontrollspeichers (CPU-Mikrocode).
    - Laden des PICO-Speichers.
    - Initialisieren des lokalen Speichers.
    - Laden der IOCs.
    - Reset der CPU, der IOCs und der ITPs.
  3. Frame 1 lokalisiert den Speicherbereich, der ausschließlich vom COS benutzt wird und aktualisiert die Systemstatus-Zeile.
  4. Steht der Autoboot-Schalter in Frame 1 auf ON, wird das Bootstrap-Programm vom Boot-Gerät geladen. Die Boot-Aktivitäten werden dann durch die in Frame 1 gesetzten Boot-Optionen beeinflusst. Steht der Autoboot-Schalter auf OFF, ist der Frame in Ruhestellung und wartet auf Eingaben. Die Nachricht COSREADY wird ausgegeben.
-



---

## Installation des Betriebssystems

---

### 7 Installation des Betriebssystems

Dieses Kapitel enthält Informationen über die Installationsbänder der TARGON /35 sowie über den Ablauf der Betriebssysteminstallation selbst.

#### 7.1 Auslieferung des Betriebssystems

Das Betriebssystem wird auf zwei verschiedenen Magnetbändern (1600 bpi) ausgeliefert:

- Band 1  
Root-Dateisystem (/) im dd-Format.
- Band 2  
/usr-Dateisystem (/usr) im cpio-Format.

Optionale Software wird auf separaten Bändern (cpio-Format) ausgeliefert.

#### 7.2 Laden des Betriebssystems

Beim Laden des Betriebssystems müssen Sie folgendermaßen vorgehen:

1. Einschalten der Stromversorgung; dazu muß der Power-Schlüssel am Frontpanel in Querstellung gebracht werden.
2. Einstellen der CE-Betriebsart; dazu muß der CE-Schlüssel am Frontpanel in Querstellung gebracht werden.
3. Einschalten der Platten- und Magnetbandlaufwerke.
4. Setzen des Plattenschalters PROTECT am Plattenlaufwerk auf OFF.

---

## Installation des Betriebssystems

---

Achtung: Durch Betätigen des Reset-Schlüssels am Frontpanel erzielen Sie nicht das gewünschte Resultat, da anschließend nur ein Reset des SSP und das Laden des COS erfolgt. Das Laden des benötigten Mikrocodes (IMPL) wird nicht vorgenommen.

Nachdem der Mikrocode geladen ist, erscheint auf der Console der Frame 1 des COS. Die Ausgabe der Systemstatus-Zeile muß folgendermaßen lauten:

-CE MODE		CPU = [0]		000FA020	COS"  "1E		*MANUAL*
----------	--	-----------	--	----------	-----------	--	----------

Jetzt kann das Root-Dateisystem eingespielt werden.

### 7.2.1 Einspielen des Root-Dateisystems

Nach Einlegen des entsprechenden Magnetbandes in das Laufwerk tätigen Sie die folgenden Eingaben:

- <CTRL> und <↖> Anwahl des Index-Frame.
- 1 Anwahl des Frame 1.
- I Reset des Systems.
- C Reset der Kanäle.
- <CTRL> und <↖> Anwahl des Index-Frame.
- 7 Anwahl Frame 7.

## Installation des Betriebssystems

### Bildschirmaufbau Frame 7 – I/O-Kanäle

<<ESC F#/OPCTL>>		-I/O CHANNEL(S) TEST-	[0] Z-> RUN/STOP
O->	OPCODE=> 00 00000000 00	XTEND BUS STATUS	--SIO INFORMATION-- 1st WORD -> 00000000 2nd WORD -> 00000000
D->	BUS DATA 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000	+BUSY +ACCEPT +NOTREADY +MCK +INVALID ADDR	
L->	LOOP (CONT.)	+OP PARITY ERR	ADDRESS ---MAIN DATA---
X->	EXECUTE BUS OPERATION	+RECEIVE ERR -MICRO ERR	
U->	UPDATE STATUS DISPLAY	-BUS UNDERRUN -BUS OVERRUN	
R->	ISSUE RESET	-SEND ERR -SIO ERR	
B->	BLOCK DATA = 00000000	-STATE 0	
F->	FILL BLOCK = 00000000	+STOP CLOCK +PRO CLOCK A +SLOW CLOCK	
M->	ALTER/DISPLAY MAIN STORE		
I->	ALTER/DISPLAY IOC#OCMEM		
-CE MODE		CPU=[0]	000FA020 COS" "1E
*MANUAL*			

In diesem Frame müssen Sie weitere Eingaben machen, um mit Hilfe des IOC das Band auf die Platte zu kopieren. Haben Sie eine Funktion ausgewählt, befindet sich der Cursor automatisch im richtigen Feld, um die entsprechenden Daten einzugeben. Die Taste <CAN> können Sie benutzen, um Eingaben, die noch nicht durch <CR> bestätigt wurden, zu löschen. Wird in einer Funktion die Eingabe von Daten in mehreren Feldern erwartet, springt durch Drücken von <CR> der Cursor von einem Feld zum anderen. Haben Sie für eine Funktion die Dateneingabe beendet, gelangen Sie mit <CAN> wieder in die Funktionsspalte. Möglicherweise müssen Sie mehrmals <CAN> betätigen, um in die Funktionsspalte zurückzukehren.

---

## Installation des Betriebssystems

---

Folgende Eingaben sind in Frame 7 vorzunehmen:

- O 1F <CR> 010C0000 <CR> <CR>
- D 010A0000 00000000 <CR>
- M <CR> 03420080 0000ABCD <CR> <CAN> <CAN>

Befindet sich der Cursor im Adreßfeld, ändern Sie durch Eingabe von <CR> XXXXXXXX in 00000000. Während der Modifikation des Hauptspeichers erscheint im Feld 7 der Systemstatus-Zeile die Anzeige MODIFIED. Nach Beendigung der Modifikation werden die Felder ADDRESS und ---MAIN DATA--- aus der Bildschirmmaske entfernt.

- X Keine weitere Eingabe erforderlich. In der Spalte XTEND BUS STATUS leuchtet das Feld +ACCEPT.

Ihre Eingaben werden durch das COS überprüft. Fehlerhafte Eingaben werden durch einen Piepton signalisiert. Das auf dem Band befindliche Root-Dateisystem wird auf die Platte kopiert. Der Kopiervorgang dauert ca. 15 Minuten.

Ist der Kopiervorgang beendet, wird das Band zurückgespult. Sie sollten sich jetzt noch einmal den Status des Hauptspeichers ansehen. Wurde das Band ohne Fehler kopiert, ist der Inhalt der Adresse Null 00000000 0000ABCD; andernfalls erscheint ein Fehlercode. Durch die Eingaben von:

- M <CR> <CR>

erhalten Sie die gewünschte Ausgabe.



## Installation des Betriebssystems

### 7.2.2 Booten des Betriebssystems

Nachdem Sie das Band mit dem Root-Dateisystem auf die Platte kopiert haben, müssen Sie Frame 1 anwählen. Um das Betriebssystem zu booten, sind folgende Eingaben in Frame 1 vorzunehmen:

**B** Laden des Bootstrap-Programms vom Boot-Gerät.

Im Feld 4 der Systemstatus-Zeile erscheint die Adresse des Boot-Programms. Wurde das Programm erfolgreich geladen, wechselt die Anzeige in Feld 7 der Systemstatus-Zeile von +BOOTING in <BOOTED>.

**Z** Starten der CPU, um das Bootstrap-Programm auszuführen.

Entweder wird das System jetzt gebootet oder es werden noch bestimmte Informationen von Ihnen gefordert. Dies ist abhängig von den Boot-Optionen, die in Frame 1 festgelegt sind.

Die Anzeige in Feld 4 der Systemstatus-Zeile wird gelöscht. Die Anzeige in Feld 8 wechselt von \*MANUAL in +RUNNING.

Wählen Sie jetzt den Terminal-Modus (Frame 0) an. Die Konfigurationskomponente erlaubt Ihnen die Konfigurierung von IOC-Kanälen und Unterkanälen, Plattenlaufwerken und ITPs, ohne einen neuen Kernel zu generieren. Nachfolgend finden Sie einige Beispiele für Konfigurationsdaten, die in Frame 0 ausgegeben werden.

Für jeden vorhandenen IOC-Kanal wird eine Zeile ausgegeben, die die Kanalnummer (Board) des IOC und seine Mikrocode-Versionsnummer enthält, z. B.:

```

I/O Configuration:
IOC 0c (version 01001 1d1)
IOC 0d (version 53001 1d1)
    
```



---

## Installation des Betriebssystems

---

Für jeden vorhandenen Unterkanal wird ebenfalls eine Zeile ausgegeben. Sie enthält die 3 Byte (Kanal/Unterkanal/Gerät) große Adresse des Geräts (iodev), den Namen des Controllers und die Controllernummer, z. B.:

```
0c0000 disk[0]
0c0100 tape[0]
0c0200 lp[0]
0d0200 lp[1]
0d0300 il[0]
```

Für jedes konfigurierte Plattenlaufwerk wird ebenfalls eine Zeile ausgegeben (aus Frame D), die den Namen der Platte, die Konfigurationsparameter und die Plattennummer (dk) enthält, z. B.:

```
disk0 (12x20x842-1-2) dk#0
disk1 (12x20x842-1-2) dk#1
disk8 (12x20x842-1-2) dk#2
disk9 (12x20x842-1-2) dk#3
```

Die letzte Zeile der ausgegebenen Informationen enthält die Liste der vorhandenen ITPs, z. B.:

```
ITP 40 41 42 43
```

Ist das System in den Single-User-Betrieb hochgefahren, ist die Systemkonsole das einzige aktive Terminal. Die Terminalparameter (tty) der Konsole müssen folgendermaßen gesetzt werden:

```
Interrupt = <CTR>C
Erase = <CTR>H
Kill = <CTR>U
```

Als letzten Schritt müssen Sie jetzt noch die Überprüfung des Root-Dateisystems durch

```
/etc/fsck /dev/disk0a
```

vornehmen. Anschließend wird der Schlüssel für den CE-Modus auf 0 gestellt.

## Installation des Betriebssystems

### 7.2.3 Laden des Dateisystems /usr

Folgende Schritte müssen Sie zum Laden von /usr vornehmen:

- Einlegen des Bandes in das Laufwerk.
- Durch Eingabe von  
/etc/load\_usr  
wird der Ladevorgang gestartet. Das Programm /etc/load\_usr arbeitet interaktiv und führt folgende Arbeiten durch:
  1. Anlegen von /usr.
  2. Kopieren des Bandinhalts auf Platte.
  3. Überprüfung des neu angelegten Dateisystems.
  4. Anlegen von /u1 und /u2.
- Nachdem der Bandinhalt auf die Platte kopiert ist (ca. 30 Minuten), kann das Band aus dem Laufwerk entnommen werden.

### 7.2.4 Anpassung systemspezifischer Dateien

Nachdem Sie das Root-Dateisystem und /usr eingespielt haben, müssen Sie die systemspezifischen Dateien – z. B. /etc/passwd, /etc/group, /usr/lib/crontab – anpassen.

---

## Installation des Betriebssystems

---

### 7.2.5 Anlegen von Sicherungskopien neuer Dateisysteme

Nach der Anpassung der systemspezifischen Dateien sollten Sie unbedingt Sicherungskopien der Dateisysteme anlegen. Dies geschieht folgendermaßen:

- Erstellen einer Sicherungskopie für das Root-Dateisystem im dd-Format:
  1. Vergewissern Sie sich, daß sich die TARGON /35 im Single-User-Mode befindet.
  2. Legen Sie ein Band in das Laufwerk ein.
  3. Geben Sie folgendes Kommando ein:

```
dd if=/dev/rdisk0a of=/dev/rmt0 bs=10k count=2000
```

Das Band muß mit 1600 bpi beschrieben werden, damit es aus Frame 7 wieder geladen werden kann.
- Erstellen einer Sicherungskopie für /usr im cpio-Format:
  1. Legen Sie ein Band in das Laufwerk.
  2. Geben Sie eins der beiden folgenden Kommandos ein:

```
cd /; find /usr -print | cpio -oaB > /dev/rmt0
```

oder

```
cd /; find /usr -cpio /dev/rmt0
```

Die Sicherung dauert ca. 45 Minuten.

---

## Installation des Betriebssystems

---

### 7.3 Generieren eines neuen Kernels

Nachdem die neuen Dateisysteme angelegt wurden, kann es erforderlich sein, einen neuen Kernel zu generieren. Die Vorgehensweise entnehmen Sie bitte dem Handbuch „TARGON /35 Systemgenerierung“.

### 7.4 Abschließende Arbeiten

Nach der Installation des neuen Root- und /usr-Dateisystems, kann der Administrator weitere Benutzerdateisysteme anlegen und die entsprechenden systemspezifischen Dateien anpassen. Anschließend bringen Sie das System in den Multi-User-Betrieb.

---



---

## Systemstart TARGON /35

---

### 8 Systemstart TARGON /35

Dieses Kapitel beschreibt das Starten der TARGON /35 und das Booten des Betriebssystems.

#### 8.1 Einschalten und Systemstart

Zuerst schalten Sie das Konsol-Terminal ein und legen die COS-Disquette in das Laufwerk. Anschließend nehmen Sie das Einschalten der Stromversorgung und den Systemstart folgendermaßen vor:

1. Power-Schlüssel am Frontpanel in Querstellung bringen.
2. CE-Schlüssel am Frontpanel in senkrechte Stellung bringen.
3. Einschalten der Bandlaufwerke.
4. Der Schalter PROTECT am Plattenlaufwerk muß auf OFF geschaltet werden und der Schalter START auf ON.

Können die jetzt anlaufenden Selbsttests nicht erfolgreich durchgeführt werden, müssen Sie die Schalter in umgekehrter Reihenfolge zurücksetzen und die Vorgänge wiederholen.

Nach dem Laden der Mikroprogramme erscheint Frame 1 auf der Konsole und Sie können das Betriebssystem booten.

**Achtung:** Durch Benutzung des RESET-Schlüssels erzielen Sie nicht das gleiche Ergebnis. Hier wird nur ein Reset des SSP und Laden des COS vorgenommen. Das Laden der Mikroprogramme wird nicht durchgeführt.

---

## Systemstart TARGON /35

---

### 8.2 Restart der TARGON /35

Ist die Stromversorgung der TARGON /35 bereits eingeschaltet, kann das System aus Frame 1 folgendermaßen gestartet werden:

<HOME> Anwahl des Index-Frame.

1 Anwahl Frame 1.

\* Auswahl IMPL MODULES.

7 ORIGINAL SYSTEM RESTART.

### 8.3 Boot-Programme

Das Betriebssystem wird beim Restart automatisch gebootet, wenn der Autoboot-Schalter in Frame 1 auf ON steht. Sie können aber auch manuell booten, indem Sie auf der Konsole das Kommando

*/etc/reboot [Optionen]*

eingeben. Eine ausführliche Beschreibung dieses Kommandos finden Sie im Anhang „Administrator-Kommandos“.

#### 8.3.1 Autoboot des Betriebssystems

Haben Sie in Frame 1 den Autoboot eingestellt, wird das Betriebssystem automatisch bei jedem Restart gebootet. Der Verlauf des Booting ist von den in Frame 1 gesetzten Boot-Optionen abhängig.

Durch folgende Eingabesequenz setzen Sie den Autoboot:

<HOME> Anwahl des Index Frame.

1 Anwahl Frame 1.

A Autoboot wird eingestellt.



## Systemstart TARGON /35

### 8.3.2 Manuelles Booten des Betriebssystems

Sie können das Betriebssystem manuell aus Frame 1 durch die folgenden Eingaben booten.

**B** Das Betriebssystem wird vom Boot-Gerät geladen.

Im Feld 4 der Systemstatus-Zeile erscheint die Adresse des Bootstrap-Programms. Nach Beendigung des Bootvorgangs wechselt die Anzeige in Feld 7 von +BOOTING in <BOOTED>.

**Z** Die CPU wird gestartet, um das Bootstrap-Programm auszuführen.

Feld 4 der Systemstatus-Zeile wird gelöscht. Die Ausgabe in Feld 8 wechselt von \*MANUAL\* in +RUNNING.

Abschließend wählen Sie Frame 0 an.

### 8.4 Shutdown und Reboot

Durch Aufruf des Programms

/etc/shutdown

wechselt der Rechner in den Single-User-Betrieb. Dieses Programm können Sie als Superuser aufrufen. Die Programme /etc/reboot und /etc/halt werden benutzt, um aus dem Single-User-Betrieb hochzufahren bzw. das System ordnungsgemäß zu stoppen.

Nach dem Booten des Systems durch das Programm /etc/reboot wird eine Überprüfung der Dateisysteme durch das Prüfprogramm fsck durchgeführt. Wird eine schwerwiegende Inkonsistenz entdeckt, gibt fsck eine entsprechende Nachricht auf der Konsole aus und terminiert. Der Single-User-Betrieb wird nicht verlassen. Die Überprüfung der Dateisysteme sollte vom Administrator dann manuell vorgenommen werden. Hat es sich bei dem fehlerhaften Dateisystem um das Root-Dateisystem gehandelt, starten Sie den Bootvorgang erneut durch

/etc/reboot -n

Eine nähere Beschreibung der in diesem Kapitel erwähnten Programme finden Sie im Anhang „Administrator-Kommandos“.



## Systemstart TARGON /35

---

### 8.5 Systemfehlermeldungen

Das Betriebssystem überprüft kontinuierlich seine internen Datenstrukturen. Beim Auftreten von Inkonsistenzen – sei es durch Hardware- oder Softwarefehler – werden Meldungen über die erkannten Fehler ausgegeben. Der Pufferspeicher der Dateisysteme und eine Kopie des physikalischen Speichers werden auf die Platte geschrieben und das System neu gebooted.

### 8.6 Verhalten bei Stromausfall

Nach einem Stromausfall sollten Sie alle Komponenten der TARGON /35, die einen Netzschalter haben, ausschalten. Ist die Stromversorgung wieder hergestellt, können Sie die Anlage und die Laufwerke wieder einschalten. Ist der Autoboot in Frame 1 aktiv, fährt die Anlage nach Stromausfällen oder Systemabstürzen selbstständig wieder hoch. Eine Überprüfung der Dateisysteme wird vorgenommen.

Ist Autoboot nicht gesetzt, müssen Sie zum Hochfahren des Systems wie im Kapitel „Einschalten und Restart“ beschrieben verfahren.

---

## Das Zwei-Welten-Konzept der TARGON /35

### 9 Das Zwei-Welten-Konzept der TARGON /35

Die Dateistruktur der TARGON /35 enthält zwei voneinander getrennte „Welten“. Eine Welt enthält den vollen Umfang von UNIX System V, die andere Welt enthält die Berkeley-Version. Diese beiden Welten werden von einem einzigen Kernel verwaltet. Dieser Kernel basiert auf dem 4.2 BSD-Kernel. Er ist kompatibel zu den Systemschnittstellen von UNIX System V und ist außerdem mit Erweiterungen der Nixdorf Computer AG ausgestattet. Dieses Zwei-Welten-Konzept erlaubt das Arbeiten entweder nur in einer Welt oder das Wechseln zwischen den beiden Umgebungen.

Die Welt, in der Sie gerade arbeiten, ist die „aktuelle Welt“. Das Login-Programm etabliert den Benutzer nach seinem Login in die in /etc/u\_universe eingetragene Welt.

#### 9.1 Kommandos, Systemschnittstellen und Spezialdateien

Zur Realisierung des Zwei-Welten-Konzepts wurden im Bereich der Kommandos, Systemschnittstellen und Spezialdateien Erweiterungen zum Standard-Betriebssystem UNIX System V implementiert. Die nachfolgende Liste enthält diese Erweiterungen, wobei der Ausdruck att für System V steht und ucb für 4.2 BSD:

att, ucb	Durch diese Kommandos wählen Sie Ihre aktuelle Welt.
In	Die Option -c stellt eine bedingte symbolische Verknüpfung her.
csymlink	Dieser System Call bewirkt eine bedingte symbolische Verknüpfung.
setuniverse	Dieser System Call setzt die Umgebung, in der die Prozesse ablaufen.
universe	Dieses Kommando zeigt Ihnen Ihre aktuelle Welt an.
u_universe	Diese Datei in /etc enthält für jeden Benutzer die Umgebung, in der er sich nach der Anmeldung befindet.



## Das Zwei-Welten-Konzept der TARGON /35

---

Eine ausführliche Beschreibung dieser Funktionen finden Sie in der entsprechenden Systemliteratur „Benutzerkommandos“ und „System-schnittstellen und Programmierung“.

### 9.2 Pipes zwischen att und ucb

Ein- und Ausgabe kann zwischen den Kommandos der beiden Welten übergeben werden. Z. B. kann ein Kommando, welches in beiden Welten vorhanden ist, in jeder Umgebung verschiedene Ausgaben erzeugen. Es ist auch möglich, daß verschiedene Optionen aufgerufen werden können. Dies wird von der aktuellen Welt, in der Sie sich befinden, bestimmt.

Beispiel:

```
find . -print | att cpio -oaB > /dev/rmt0
```

Diese Form der Kommandoeingabe erlaubt Ihnen, temporär die ucb-Umgebung zu verlassen, das System V-Kommando cpio aufzurufen und dann in die ucb-Welt zurückzukehren. Außerdem wird die Ausgabe des ucb-Kommandos find in die att-Umgebung an cpio übergeben. Generierung von Dateinamen und Umleitung der Standardausgabe wird in der aktuellen Welt ausgeführt.

## Das Zwei-Welten-Konzept der TARGON /35

### 9.3 Dateistrukturen

Zur Implementierung der zwei Welten war die Erweiterung des In-Kommandos um die Option `-c` erforderlich. Diese Option definiert eine bedingte symbolische Verknüpfung für Dateien oder Verzeichnisse, die entweder in beiden Welten oder nur in einer vorhanden sind

Die folgende Tabelle enthält alle `/` und `/usr` Verzeichnisse, die bedingt symbolisch verknüpft sind.

<b>Bedingte symbolische Verknüpfungen</b>		
Name	Verknüpfte Verzeichnisse, wenn aktuelle Welt:	
	<b>att</b>	<b>ucb</b>
<code>/bin</code>	<code>/.attbin</code>	<code>/.ucbbin</code>
<code>/lib</code>	<code>/.attlib</code>	<code>/.ucblib</code>
<code>/usr/bin</code>	<code>/usr/.attbin</code>	<code>/usr/.ucbbin</code>
<code>/usr/lib</code>	<code>/usr/.attlib</code>	<code>/usr/.ucblib</code>
<code>/usr/man</code>	<code>/usr/.attman</code>	<code>/usr/.ucbman</code>
<code>/usr/include</code>	<code>/usr/.attinclude</code>	<code>/usr/.ucbinclude</code>
<code>/usr/ucb</code>	<code>/usr/.attucb</code>	<code>/usr/.ucbucb</code>

Existiert eine solche Verknüpfung, wird das der aktuellen Welt entsprechende Verzeichnis angesprochen. Befinden Sie sich z. B. im `att`-Universum und sprechen das Verzeichnis `/bin` an, dann wird tatsächlich das Verzeichnis `/.attbin` betrachtet.





## Initialisierungsprogramme

### 10 Initialisierungsprogramme

Aufgrund der unterschiedlichen Arbeitsweise der System V- und Berkeley-Versionen von init, getty und login finden Sie in diesem Kapitel die Gegensätze beschrieben.

Das Betriebssystem der TARGON /35 kann jeweils nur eine Version dieses Programmpakets nutzen. Die TARGON /35 wird mit der System V-Version dieser Programme installiert. Nach der Installation kann der Systemadministrator das System V-Paket durch das Berkeley-Paket ersetzen. Die Verzeichnisse /etc/INIT/4.2 und /etc/INIT/5.0 enthalten jeweils die entsprechenden Versionen von init, getty und login.

Nachfolgend finden Sie eine kurze Zusammenfassung der Aufgaben, die diese Programme ausführen. Eine detaillierte Beschreibung der Arbeitsweise der System V-Programme und finden Sie im Anhang „Administrator-Kommandos“.

#### ■ /etc/init

wird als letzter Schritt des Bootvorgangs gestartet um:

- Start-Scripts auszuführen,
- das System in den Multi-User-Betrieb zu bringen und
- ständig getty-Prozesse zu erzeugen.

4.2 BSD-init benutzt die Dateien inittab, rc, ttys, utmp und wtmp.

System V-init benutzt die Dateien inittab, bcheckrc, brc, rc, utmp und wtmp.

#### ■ /etc/telinit

ist mit /etc/init verknüpft und steuert die Aktionen von init. Es wird mit einem einzelnen Argument aufgerufen und veranlaßt init über den Systemaufruf kill die entsprechenden Aktionen auszuführen. Telinit wird aufgerufen, um:

- init zu veranlassen, das System in einen der Run-Level 0-6 zu setzen,
- init zu veranlassen, nur die Einträge in der Datei /etc/inittab zu bearbeiten, in denen der Run-Level a, b oder c gesetzt ist,

---

## Initialisierungsprogramme

---

- init zu veranlassen, die Datei inittab wieder zu lesen und
- init zu veranlassen in den Single-User-Modus (s oder S) umzuschalten. Wird dieser Wechsel durchgeführt, ist die virtuelle Systemkonsole /dev/syscon das Terminal, von dem das Kommando abgesetzt wurde.

### ■ **/etc/getty**

wird von init aufgerufen um:

- Terminalleitungen zu eröffnen und deren Modi einzustellen,
- die Login-Aufforderung auszugeben und
- einen Login-Prozeß für die Benutzer anzustoßen.

4.2 BSD-getty benutzt die Dateien gettytab, utmp und wtmp.

System V-getty benutzt die Dateien gettydefs, utmp und wtmp.

### ■ **/bin/login**

wird von getty aufgerufen um:

- die Paßwortprüfungen für die Benutzer durchzuführen,
- das Login-Universum der Benutzer einzustellen,
- zu prüfen, ob Post vorhanden ist und
- die Shell des Benutzers zu starten.

4.2 BSD-login benutzt die Dateien gettydefs, ttys, ttytype, u\_ttytype, u\_universe, utmp und wtmp.

System V-login benutzt die Dateien ttys, ttytype, u\_universe, utmp und wtmp.

### ■ **/etc/shutdown**

versetzt das System in den Single-User-Betrieb.

4.2 BSD-shutdown und System V-shutdown benutzen die Datei wtmp.



## Initialisierungsprogramme

### 10.1 Terminal-, Login- und Accounting-Dateien

Das Management zweier UNIX-Welten auf einem Rechner erfordert die Verwaltung verschiedener administrativer Dateien. Die Verwaltung dieser Dateien ist weder aufwendig noch langwierig. Das System TARGON /35 wird mit Standardversionen dieser Dateien ausgeliefert. Der Administrator muß lediglich terminal- und leitungsspezifische (TERM) Angaben sowie Login-Informationen eintragen.

Nachfolgend beschreiben wir Ihnen die unterschiedliche Nutzung dieser administrativen Dateien unter 4.2 BSD und System V.

#### ■ /etc/ttys

versorgt 4.2 BSD-init mit folgenden Informationen:

- Terminalleitungen, für die ständig getty-Prozesse erzeugt werden sollen.
- Suchmuster für Terminalmodi in der Datei gettytab.

System V-login und 4.2 BSD-login benutzen diese Datei für das ucb-Login-Accounting (/etc/utmp).

**Achtung:** Ist diese Datei nicht vorhanden, werden keine getty-Prozesse für 4.2 BSD-init erzeugt. Außerdem findet für System V-init kein ucb-Accounting statt.

#### ■ /etc/inittab

enthält Einträge, die von System V-init folgendermaßen benutzt werden:

- Ausführung bestimmter Arbeiten während des Bootens und
- Erzeugung von getty-Prozessen.

4.2 BSD-init benutzt diese Datei, um in der att-Welt Accounting durchzuführen.

System V-who benutzt diese Datei ebenfalls zur Erfüllung diverser Aufgaben in beiden Versionen von init.

---

## Initialisierungsprogramme

---

### ■ **/dev/init**

ist eine zeichenorientierte Gerätedatei, die dazu benutzt wird, dem init-Prozeß (PID = 1) Nachrichten zu übergeben. Die Hauptaufgabe dieser Datei besteht darin, unter dem Login-Accounting das Accounting von Remote-Prozessen zu überwachen.

Versucht ein Benutzer sich remote mittels rlogin oder telnet einzuloggen, wird die Datei /dev/init von Accounting Dämon-Prozessen eröffnet. /dev/init sendet dann eine Nachricht an den init-Prozeß, um das Anmelden auf einer bestimmten Terminalleitung (Pseudo-Tty) zu ermöglichen. Die Anmeldung wird dann zugelassen, wenn für das Terminal ein Eintrag in /etc/inittab vorhanden ist und kein anderer Benutzer zur Zeit an diesem Terminal arbeitet.

```
rp5:r:remote:daemon ttyp5
```

Dieser Eintrag in inittab bedeutet, daß eine Remote-Anmeldung für /dev/tty5 möglich ist. Das Accounting findet dann unter der ID rp5 in utmp statt.

Achtung: Der Eintrag in /etc/inittab spezifiziert nicht die Art der Remote-Anmeldungen. Er bewirkt lediglich, daß /dev/init eröffnet wird und die Nachricht sendet, daß eine Pseudo-Terminalleitung eröffnet werden soll.

Der Systemadministrator muß folgende Punkte beachten:

1. Der Eigentümer der Datei /dev/init sollte Root sein. Außer für den Eigentümer sind keine weiteren Zugriffsrechte zugelassen (600).
2. Für jedes Terminal, auf das remote zugegriffen werden soll, muß ein Eintrag in /etc/inittab vorhanden sein.
3. Wurde /dev/init gelöscht, ist keine Remote-Anmeldung möglich, bis die Datei wieder angelegt wird.

### ■ **/etc/gettytab**

enthält Terminaleinstellungen und die Login-Nachricht.

4.2 BSD-getty benutzt diese Datei, um die Modi der Terminals zu setzen.

## Initialisierungsprogramme

### ■ **/etc/gettydefs**

enthält Terminaleinstellungen und die Login-Nachricht.

System V-getty benutzt diese Datei, um die Modi der Terminals zu setzen.

4.2 BSD-getty benutzt diese Datei, um Terminaleinstellungen für die Benutzer vorzunehmen, deren Standardumgebung nach der Anmeldung die att-Welt ist.

### ■ **/etc/u\_ttytype**

enthält Terminaleinstellungen.

4.2 BSD-login benutzt diese Datei, um die Terminaleinstellungen für System V aus gettydefs zu erfahren.

### ■ **/etc/ttytype**

enthält die Terminalleitung für die Variable TERM.

System V- und 4.2 BSD-login benutzen diese Datei zum Setzen der Variable TERM

- **/etc/u\_universe** enthält Einträge, die die Standardumgebung der Benutzer nach der Anmeldung definieren. Diese Datei wird von System V- und 4.2 BSD-login gleichermaßen benutzt.

Achtung: Für jeden Eintrag in /etc/passwd sollte ein Eintrag in /etc/u\_universe vorhanden sein.

In folgenden Fällen befindet sich ein Benutzer nach der Anmeldung automatisch in der ucb-Welt:

1. In der Datei /etc/u\_universe ist kein Eintrag für den entsprechenden Benutzer vorhanden.
2. Der Eintrag in /etc/u\_universe lautet weder ucb noch att.
3. Die Datei /etc/u\_universe ist nicht vorhanden.

---

## Initialisierungsprogramme

---

### — **/etc/utmp**

enthält in beiden Welten Accounting-Informationen über die angemeldeten Benutzer. Zusätzlich sind in der Datei utmp unter System V noch folgende Informationen enthalten:

- Zeitpunkt, an dem die Anlage zuletzt gebootet wurde und
- wieviele getty-Prozesse bereit sind, login-Prozesse zu werden.

Da Programme in beiden Welten die Datei /etc/utmp nutzen, wurde eine bedingte symbolische Verknüpfung auf /etc.attutmp und /etc.ucbutmp hergestellt.

### — **/etc/wtmp**

enthält kumulierte Abrechnungssätze (System V).

Da wtmp im Verzeichnis /etc angelegt ist, wurde eine bedingte symbolische Verknüpfung auf /etc.attwtmp hergestellt.

### — **/usr/adm/wtmp**

enthält kumulierte Abrechnungssätze (4.2 BSD).

Da wtmp im Verzeichnis /usr/adm angelegt ist, wurde eine symbolische Verknüpfung auf /etc.ucbwtmp hergestellt.

### — **/etc/security**

steuert die Sicherungsmaßnahmen, die von login (ucb und att) durchgeführt werden müssen.

### — **/etc/cshinit.{att oder ucb}**

erlaubt dem Systemadministrator, den Benutzern von csh oder tcsh Standard-Umgebungsvariablen zur Verfügung zu stellen.

## Initialisierungsprogramme

### 10.2 rc-Dateien

Die Datei `/etc/rc` enthält Prozeduren, die von `init` ausgeführt werden, bevor das System in den Multi-User-Betrieb hochgefahren wird. System V-`init` führt vor `/etc/rc` erst `/etc/bcheckrc` und `/etc/brc` aus. Die 4.2 BSD- und System V-`rc`-Dateien sind unterschiedlich und können ohne entsprechende Änderungen nicht ausgetauscht werden.

#### 10.2.1 System V-init

■ **`/etc/bcheckrc`**

wird einmal – vor `brc` – ausgeführt und initiiert folgende Arbeiten:

- Vorbereitung des Systems für den Multi-User-Betrieb und
- Ausführung des Prüfprogramms `fsck`.

■ **`/etc/brc`**

wird einmal – nach `bcheckrc` und vor `rc` – ausgeführt und initiiert das Laden des Mikrocodes in die ITPs.

■ **`/etc/rc`**

wird als letzte der Initialisierungsdateien ausgeführt. Folgende Arbeiten werden durchgeführt:

- Mounten der Dateisysteme,
- Starten der Systemdämonen,
- Starten des Spooler-Schedulers,
- Starten des Accounting und des Fehler-Loggings sowie des Loggings für Systemaktivitäten,
- Leeren von `/tmp` und
- Hochfahren in den Multi-User-Betrieb.

Bei späteren Änderungen des Run-Levels gibt `rc` eine Nachricht aus, daß der Wechsel in den Multi-User-Modus durch Eingabe von `telinit 2` erfolgt.

## Initialisierungsprogramme

---

### 10.2.2 4.2 BSD-init

#### **/etc/rc**

enthält folgende Initialisierungsprozeduren:

- Ausführung des Prüfprogramms fsck,
- Start der Systemdämonen,
- Einhängen der Dateisysteme,
- Laden des Mikrocodes in die ITPs,
- Start des Spooler-Schedulers,
- Start des Accounting und des Fehler-Loggings sowie des Loggings für Systemaktivitäten,
- Leeren von /tmp und
- Hochfahren in den Multi-User-Betrieb.

## Initialisierungsprogramme

### 10.3 Login-Dateien für Benutzer

Manchmal steht der Systemadministrator vor der Notwendigkeit, Dateien reparieren zu müssen, die von den `init-`, `getty-` und `login-` Prozessen benutzt werden, um die Anmeldung der Benutzer ausführen zu können. In den folgenden Abschnitten finden Sie Informationen über die Reparatur solcher Dateien bzw. über das Kreieren bzw. Löschen von `getty-` Prozessen.

#### 10.3.1 Reparaturen in `/etc/inittab` und `/etc/utmp`

Die Datei `/etc/inittab` muß für jedes Terminal einen Eintrag enthalten. Ist dies nicht der Fall, ist es möglich, daß die Anmeldung eines Benutzers abgewiesen wird.

Aufgrund einer fehlerhaften `/etc/utmp`-Datei kann es zu folgender Meldung kommen:

```
No utmp entry. You must exec "login" from the lowest level "sh".
```

Wird eine Anmeldung zurückgewiesen, haben Sie die folgenden Möglichkeiten diesem Fehler abzuhelpfen:

1. Fahren Sie das System in den Single-User-Betrieb und überprüfen Sie die Datei `/etc/inittab`.
2. Löschen Sie `utmp` (`att`) mit folgendem Kommando:  

```
# rm /etc.attutmp
```
3. Booten Sie das System.

Die `/etc/utmp`-Dateien können Sie – nachdem Sie das System in den Single-User-Betrieb gebracht haben – folgendermaßen reparieren:

1. Löschen Sie den Inhalt der Dateien in beiden Welten durch die folgenden Kommandos:  

```
# cp /dev/null /etc.attutmp
# cp /dev/null /etc.ucbutmp
```

---

## Initialisierungsprogramme

---

- Überzeugen Sie sich von der Existenz der bedingten symbolischen Verknüpfung durch Eingabe von:

```
# ls -l /etc/utmp
```

Datei und Verknüpfung sollte folgendermaßen angezeigt werden:

```
utmp -> att:.attutmp ucb:.ucbutmp
```

- Ist die Datei oder die Verknüpfung nicht vorhanden, können Sie sie durch folgende Kommandos wieder anlegen:

```
# cd /etc
```

```
# rm -f utmp
```

```
# ln -c att:.attutmp ucb:.ucbutmp utmp
```

- Booten der Anlage und Hochfahren in den Multi-User-Betrieb.

### 10.3.2 Reparaturen in wtmp

Eine fehlerhafte oder nicht vorhandene wtmp-Datei behindert nicht die Ausführungen von init, getty, login oder shutdown. Diese Prozesse können jedoch die kumulierten Abrechnungssätze nicht in wtmp ablegen, solange die Datei nicht repariert ist. Um die Datei und die zugehörigen Verknüpfungen wieder anzulegen, geben Sie die folgenden Kommandos ein:

```
# cp /dev/null /etc/.attwtmp
```

```
# cp /dev/null /etc/.ucbwtmp
```

```
# cd /etc
```

```
# ln -c att:.attwtmp wtmp
```

```
# cd /usr/adm
```

```
# ln -s /etc/.ucbwtmp wtmp
```

Zur Ausführung dieser Kommandos ist es nicht zwingend notwendig, das System in den Single-User-Modus zu bringen.



## Initialisierungsprogramme

### 10.3.3 Getty-Prozesse

Um getty-Prozesse zu entfernen oder anzulegen, müssen Sie die Dateien /etc/inittab und /etc/ttys entsprechend editieren. Anschließend veranlassen Sie init durch die Eingabe von

```
# kill -HUP 1
```

die Datei ttys zu lesen (ucb) oder durch

```
# init q
```

die Datei /etc/inittab zu lesen (att).

### 10.4 Login-Sicherheitsmaßnahmen

Das Programm /bin/login stellt in beiden Welten mehrere Sicherheitsmaßnahmen zur Verfügung:

- Paßwortänderung in gewissen Zeitabständen (nur System V).
- 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.

---

## Initialisierungsprogramme

---

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 (s. auch Anhang „Dateiformate“: <code>passwd</code> ).
PASSREQ	Für jeden Benutzer ist ein Paßwort vorgeschrieben.
LOG_TO_FILE	Fehler beim Login werden in <code>/usradm/badlog</code> aufgezeichnet.
THREE_TRY	Die Anzahl der nicht erfolgreichen Anmeldungen wird auf drei begrenzt.
DIALPASS	Für Wählverbindungen wird ein zusätzliches Paßwort verlangt.

Das Paßwort für Wählverbindungen wird verschlüsselt und in der Datei `/etc/d_passwd` abgelegt. Ist diese Datei nicht vorhanden, findet keine Überprüfung des Paßworts für Wählverbindungen statt.

### 10.5 Einrichten der Umgebung für `csh` und `tcsh`

Das Programm `login` stellt einen einfachen Mechanismus zur Verfügung, um den Benutzern von `csh` und `tcsh` Umgebungsvariablen zu setzen und zwar die Dateien `/etc/cshinit.att` und `/etc/cshinit.ucb`. Durch Einträge in diesen Dateien können Sie folgendes erreichen:

- Überschreiben der Variable `PATH`, um andere wichtige Verzeichnisse zu erfassen.
- Setzen des Zeitzonen-Parameters (`TZ`) für alle Benutzer.
- Setzen von lokal genutzten Variablen.

Benutzer der `ucb`-Welt entnehmen Umgebungsvariablen aus `/etc/cshinit.ucb`; analog dazu entnehmen die Benutzer der `att`-Welt ihre Umgebungsvariablen der Datei `/etc/cshinit.att`. Ist der Inhalt beider Dateien identisch, können sie verknüpft werden. Eigentümer dieser Datei sollte `Root` sein, Leseerlaubnis für alle Benutzer ist notwendig.

## Initialisierungsprogramme

Merke: Umgebungsvariablen, die ein Benutzer in seinen persönlichen Dateien .login oder .cshrc setzt, überschreiben die Variablen in /etc/cshinit.{att oder ucb}. Bei der Ausführung der System V-Version von init, getty und login werden Umgebungsvariablen, die in einer login-Kommandozeile gesetzt sind, durch Einträge in cshinit.{ucb oder att} überschrieben.

Jeder Eintrag in /etc/cshinit muß durch ein Zeilenende-Zeichen beendet werden und hat folgende Form:

*VARNAME=Zeichenkette*

Analog zu der Kommandoeingabe in der Shell können Sie durch die Eingabe eines Backslash (\) einen Eintrag über mehrere Zeilen schreiben. Anführungszeichen und Gleichheitszeichen werden entfernt. Diese Eigenschaft können Sie jeweils durch einen vorangestellten Backslash unwirksam machen.

### 10.6

#### Wechseln der Initialisierungsprogramm-Versionen

Die TARGON /35 benutzt standardmäßig die System V-Version der Initialisierungsprogramme. Die 4.2 BSD-Versionen von init, getty, login und shutdown werden mit ausgeliefert und sind im Verzeichnis /etc/INIT/4.2 abgelegt. Kopien der System V-Versionen finden Sie im Verzeichnis /etc/INIT/5.0. Die beiden Verzeichnisse enthalten ebenfalls Muster für rc-Dateien. Da die beiden Versionen der Kommandos unterschiedlich arbeiten, dürfen sie nicht untereinander ausgetauscht werden. Sie können also beispielsweise nicht das System V-login zusammen mit 4.2 BSD-init und -login nutzen.

Zur Installation der beiden unterschiedlichen Pakete (init, getty, login und shutdown) stehen Ihnen zwei Shell-Prozeduren zur Verfügung. Mit der Prozedur inst.att installieren Sie die Dateien aus /etc/INIT/5.0 und inst.ucb steht für die Installation der Dateien aus /etc/INIT/4.2 zur Verfügung.

© „Weitergabe sowie Vervielfältigung dieser Unterlage...“  
Zusammenhang verpflichtet zu Schadenersatz. Alle Rechte für den Fall der Patenterteilung oder Gebrauchsmustereintragung vorbehalten.“

---

## Initialisierungsprogramme

---

Wollen Sie wahlweise eines der Pakete nutzen, ist folgendes zu beachten:

1. Das System V-Paket ist das Standardpaket der TARGON /35. Die Dateien unter /etc/INIT/5.0 sind in den Verzeichnissen /etc und /bin installiert.
2. Das 4.2 BSD-Paket finden Sie – zusammen mit allen Dateien, die Sie zur Implementierung benötigen – im Verzeichnis /etc/INIT/4.2.
3. Bei ständiger Aktualisierung der Dateien (z. B. rc), die zu den beiden Paketen gehören, werden Sie feststellen, dass der Wechsel der Programmpakete sehr einfach ist.
4. Inst.all und inst.ucb überprüfen die aktuell in /etc und /bin installierten Dateien auf Änderungen. Abgeänderte Dateien, die überschrieben werden, werden im Verzeichnis /etc/INIT/save gesichert. Anschließend wird das gewünschte Paket installiert.

**Bitte sichern Sie die ursprünglich ausgelieferte Software, damit jederzeit eine Rekonstruktion des Urzustands möglich ist.**

5. Um die Installation durchzuführen, müssen Sie das System in den Single-User-Betrieb fahren. Sofort nach Beendigung der Installation ist das System wieder zu booten, um eine sichere Installation zu gewährleisten.

### 10.6.1 Vorgehensweise beim Austausch des System V-Pakets

1. Vergewissern Sie sich, daß in der Datei /etc/INIT/4.2/rc Einträge für alle Aufgaben vorhanden sind, die unter System V durch die Dateien
  - /etc/bcheckrc
  - /etc/brc
  - /etc/rcinitiiert werden.
2. Die Einträge in /etc/inittab müssen mit /etc/INIT/4.2/ttys übereinstimmen.

---

## Initialisierungsprogramme

---

3. Die Datei `/etc/INIT/4.2/gettytab` muß vorhanden sein und muß die Standardwerte für die Terminals enthalten.
4. `/etc/INIT/4.2/u_ttytype` muß mit `/etc/inittab` bezüglich der Einträge in `/etc/gettydefs` übereinstimmen.
5. `/etc/INIT/4.2/shutdown5` muß alle Dämonprozesse unter System V erreichen, die ordnungsgemäß terminiert werden müssen.

### 10.6.2 Vorgehensweise beim Austausch des 4.2 BSD-Pakets

1. Überprüfen Sie, ob die Dateien `brc`, `bcheckrc` und `rc` mit der Datei `rc` unter 4.2 BSD übereinstimmen.
2. `/etc/INIT/5.0/inittab` muß mit `/etc/ttys` übereinstimmen.
3. Überprüfen Sie, ob in `/etc/INIT/5.0/gettydefs` die entsprechenden Einträge für Übertragungsgeschwindigkeit usw. vorhanden sind. Die Datei `/etc/gettytab` ist unter 4.2 BSD das Gegenstück zu `gettydefs`. Sie können sich in `gettytab` einen Überblick verschaffen, welche Einträge vorhanden sein müssen.
4. Überprüfen Sie, ob alle Änderungen, die in `/etc/shutdown5` gemacht wurden, auch in `/etc/INIT/5.0/shutdown` eingetragen sind.

Diese Informationen bezüglich der beiden Pakete finden Sie – in englischer Sprache – auch auf Ihrer TARGON /35 in der Datei `/etc/INIT/README`.

---



## Laufende Arbeiten der Systemverwaltung

### 11 Laufende Arbeiten der 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.

#### 11.1 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 (s. Benutzerkommando `chmod`). Bei Shell-Prozeduren funktioniert dieser Mechanismus nicht.

Erkennbar sind diese Programme mit dem Kommando `ls -l`:

```
-rws  rws  rwx
```

Bei Ausführung erhält der Benutzer die Rechte der Gruppe

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.

---

## Laufende Arbeiten der Systemverwaltung

---

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 ergeben.
- 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.

### 11.2 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. Daher kommt diesem Kapitel eine große Bedeutung zu.

#### 11.2.1 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 sollte eine inkrementelle Sicherung aller Benutzer-Dateisysteme auf ein Sicherungsmedium durchgeführt werden. Diese Medien sollten bis zur nächsten Vollsicherung aufbewahrt werden, bevor sie erneut zur Sicherung benutzt werden. Als Sicherungsmedium kann die Platte eines anderen Laufwerks oder ein Magnetband fungieren.



## Laufende Arbeiten der Systemverwaltung

- **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.

### 11.2.2 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.
- **volcopy**  
Volcopy wird zur physikalischen Sicherung von Dateisystemen benutzt. (Siehe auch Anhang „Administrator-Kommandos“.)
- **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. Mit tar erfolgt die Sicherung schneller als mit cpio, es können jedoch keine Gerätedateien gesichert werden.

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

---

## Laufende Arbeiten der Systemverwaltung

---

### 11.3 Verkleinern von Verzeichnissen

Verzeichnisse, deren Größe 4 KByte (= 256 Einträge) übersteigt, sind – aufgrund des indirekten Zugriffs – sehr ineffizient. 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
```

## Laufende Arbeiten der Systemverwaltung

### 11.4 Reinigen der administrativen Log-Dateien

Die meisten der folgenden Dateien werden beim Booten durch Einträge in /etc/rc automatisch zum Schreiben eröffnet. Die Dateien werden sehr schnell 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 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.

---

## Laufende Arbeiten der Systemverwaltung

---

### 11.5 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 angemeldeten Benutzer zu erreichen wird das Kommando `wall` (write all) benutzt. `wall` sollte vom Super-User 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.

### 11.6 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 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.

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

Beispiel:

```
find / -mtime +30 -atime +30 -print > verwaltung
```

## Laufende Arbeiten der Systemverwaltung

Die Aufzeichnungen in der Datei `verwaltung` enthalten alle die Dateien, bei denen in den letzten 30 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
```

Wollen Sie die Verzeichnisse mehrerer Benutzer auf diese Weise in ein anderes Dateisystem verschieben, sollten Sie darauf achten, daß Gruppenzusammengehörigkeiten gewahrt bleiben. Außerdem müssen Sie eine Gruppenverschiebung immer mit einem `cpio`-Kommando vornehmen, da sonst Dateiverknüpfungen gelöst werden.

## Laufende Arbeiten der Systemverwaltung

---

### 11.7 Definition der Standard-Benutzerumgebung (/etc/profile)

Nach dem Anmelden eines Benutzers wird sofort nach dem Starten der Shell das Shellsript /etc/profile ausgeführt, bevor die benutzereigene Umgebung durch .profile erzeugt wird. Das erlaubt dem Systemverwalter durch entsprechende Einträge in /etc/profile eine Standardumgebung für alle Benutzer zu schaffen (z. B. Ausführen von umask, Setzen von Shell-Variablen usw.).

### 11.8 Eintragen neuer Benutzer

Es ist sehr einfach, das System TARGON /35 für neue Benutzer zugänglich zu machen. Folgende Informationen sollten Sie vorher einholen:

- Name des Benutzers (sollte zur besseren Übersicht in das Feld für die Accounting-Information eingetragen werden).
- Login-Name (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).

Auf Grund dieser Informationen können Sie die erforderlichen Einträge in die Dateien /etc/passwd und /etc/group vornehmen.

## Laufende Arbeiten der Systemverwaltung

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

1. Editieren eines Eintrags in `/etc/passwd` für den neuen Benutzer.
2. Editieren eines Eintrags in `/etc/group`.
3. Editieren eines Eintrags in `/etc/u_universe`.
4. Anlegen des in `/etc/passwd` eingetragenen Login-Verzeichnisses.
5. Ändern der Benutzer- und Gruppen-ID des soeben angelegten Login-Verzeichnisses für den neuen Benutzer (`chown`, `chgrp`).
6. Melden Sie sich zur Kontrolle in dem neuen Verzeichnis an.
7. Eingeben eines Paßwortes für den neuen Benutzer.
8. 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.

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.

---

## Laufende Arbeiten der Systemverwaltung

---

### 11.9 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.
  - Accounting
  - Verwaltung von Dateisystemen
  - Langlaufende, vom Benutzer geschriebene Shell-Prozeduren, die das Kommando su benutzen. Z. B.:  
**su -userx userx\_shell arg ...**
- Sperren der Programme im Verzeichnis /usr/games während bestimmter Zeiten.



---

## Laufende Arbeiten der Systemverwaltung

---

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
  - 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 reduzieren usw.
    - Zuweisung von Betriebsmitteln.
-



## Das Spoolsystem

### 12 Das Spoolsystem

Dieses Kapitel beinhaltet einen Überblick über das Spoolsystem sowie die Ausführung geschützter Funktionen.

Das Spool-System ist ein Kommandosystem, das verschiedene Spooler-Funktionen unter dem TARGON /35-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.

#### 12.1 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 Ip-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 Spoolsystem

---

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.

### 12.2 Kommandos

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

#### 12.2.1 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 Spoolsystem

### 12.2.2 Administrator-Kommandos

Jedes Spoolsystem 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 (Anhang „Administrator-Kommandos“). Normalerweise gehören alle Spooler-Dateien und -Kommandos dem Benutzer lp, nur lpadmin und lpsched gehören dem Superuser.

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

## Das Spoolsystem

---

### 12.3 Die Installation des Spoolsystems

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.

#### 12.3.1 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 Spoolsystem

---

### 12.4 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.

#### 12.4.1 Druckerkonfiguration

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

##### 12.4.1.1 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 lp beschrieben werden kann.

---

## Das Spoolsystem

---

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*).

### 12.4.1.2 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 Druckerstatistika – 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 Spoolsystem

### 12.4.1.3 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 hp ist:

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

2. Konfigurieren Sie einen Drucker mit dem Namen dr2, dessen Pfadname /dev/tty22 und dessen Schnittstellenprogramm eine Variation des prx-Programms ist. Die Gerätedatei /dev/tty22p ist gleichzeitig ein Terminal:

```
$ cp /usr/spool/lp/model/prx xxx
< xxx editieren >
$ /usr/lib/lpadmin -pdr2 -v/dev/tty22p -ixxx -l
```

3. Legen Sie einen neuen Drucker mit dem Namen dr3 und dem Pfadnamen /dev/tty23p 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/tty23p -edr2 -ccl1
```

### 12.4.2 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*).

---

## Das Spoolsystem

---

- Änderung der Druckerspezifizierung (-h oder -l).
- Zuordnung der Drucker zu einer bestehenden oder neuen Klasse (-c*Klasse*).
- Entfernung des angegebenen Druckers aus dem Spool-System (-x*Drucker*). Ist dieser Drucker das einzige Mitglied einer Klasse, wird die Klasse ebenfalls gelöscht.
- Entfernung von Druckern aus einer bestehenden Klasse (-r*Klasse*). 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.

### 12.4.3 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/tty24p und fügen Sie ihn in eine neue Klasse mit dem Namen cl2 ein:

```
$ /usr/lib/lpadmin -pdr2 -mprx -v/dev/tty24p -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
```

## Das Spoolsystem

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 -rc11
```

```
$ /usr/lib/lpadmin -pdr3 -rc11
```

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
```

### 12.4.4 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
```

---

## Das Spoolsystem

---

### 12.4.5 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.

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.

## Das Spoolsystem

### 12.5 Der Scheduler

In diesem Abschnitt 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.

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

#### 12.5.1 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`

## Das Spoolsystem

---

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.

### 12.5.2 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 Spoolsystem

### 12.6 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:

Drucker-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 Spoolsystem

---

### 12.6.1 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

### 12.6.2 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 Spoolsystem

### 12.6.3 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 verloren gehen. Wenn ein arbeitender Drucker den Status disable erhält, wird das Schnittstellenprogramm durch Signal 15 beendet.

---

## Das Spoolsystem

---

### 12.7 Einsatz von Geräten und Terminals als Drucker

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

#### 12.7.1 Geräte

Als Beispiel für den Einsatz von Geräten als Drucker richten Sie die Terminalleitung 15 (VT100) als Drucker xyz ein. Führen Sie als Super-user 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/tty15
$ chmod 600 /dev/tty15
```

2. Ändern Sie /etc/inittab so, daß tty15 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:off:
```

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 tty15 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/tty15 -mprx
```

---

## Das Spoolsystem

---

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.

### 12.7.2 Terminals

Terminals können ebenfalls als Drucker behandelt werden. Um z. B. ein VT100-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
```

## Das Spoolsystem

---

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/ttyi02) mit dem Drucker abc:

```
$ /usr/lib/lpadmin -pabc -v/dev/ttyi02
```

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

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: I/O-Fehler-Codes

### A1 I/O-Fehler-Codes

Die TARGON /35 protokolliert mit Hilfe von errpt I/O-Fehler, die durch den errdemon-Mechanismus ermittelt werden. Im folgenden Beispiel finden Sie die Aufzeichnung eines IOCDISK-Fehlers. Diese Reports werden von errpt generiert.

IOCDISK Error Logged On	Wed May 14 10:34:45 1986
Physical Device	
Channel	0x0c
Sub-Channel	0x00
Device	0x02
Retry Count	8
Error Diagnosis	Unrecovered
Interrupt Status	0x100601
Type of Transfer	Write
I/O Type	Buffered
Transfer Size in Bytes	0
Block Number	0
Cylinder	0
Head	0
Sector	0
Statistics On Device To Date:	
R/W Operations	0
Other Operations	1
Unrecorded Errors	0

Diese, von errpt generierten Aufzeichnungen sind selbsterklärend. Lediglich der „Interrupt Status“ (in unserem Beispiel 0x100601) muß interpretiert werden, um festzustellen, welcher Fehler die Ursache des Interrupts war.

© „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 Gebrauchsmustereintragung vorbehalten.“

---

## Anhang 1: I/O-Fehler-Codes

---

Der Interrupt Status ist ein drei Byte langer Wert, der hexadezimal ausgegeben wird (0x = hexadezimale Ausgabe). Die ersten beiden Ziffern beinhalten den „Ursache-Code“, gefolgt von den beiden Ziffern des „Fehler-Codes“. Die letzten beiden Ziffern sind das Status-Byte. Diese Ausgabe lautet immer 01 und bedeutet, daß ein Fehler aufgetreten ist (00 = kein Fehler).

Nachfolgend finden Sie eine Liste der Fehler-Codes. Diese Codes verweisen auf das E/A-Gerät, welches Ursache für den Stillstand ist.

00	No Error
01	IOC Crash
02	Bad SIO
03	Bad Command IOCB
04	Bad DCHAIN IOCB
05	system bus Error
06	Disk Error
07	Tape Error
08	Printer Error
09	Ethernet Error
0a	DR11W Error (Graphics)
0b	Arpanet Error
0c	Versatec

Fehler in den I/O-Geräten 00, 01 und 05 produzieren keinen Ursache-Code. Fehler im Gerätetyp 02 (Bad SIO) produzieren die folgenden Ursache-Codes:

01	Transmission error, message
02	Bad function field, message
03	Address not word aligned, message
04	Duplicate job, message

## Anhang 1: I/O-Fehler-Codes

Fehler im Gerätetyp 03 (Bad Command IOCB) produzieren die folgenden Ursache-Codes:

- 01 Can't read the IOCB, iocbfill
- 02 The next IOCB pointer is bad, next Bad command number, mid
- 04 Bad sub-channel number, subchan
- 05 Bad device number, disk (Disk)
- 06 Bad command number, disk (Disk)
- 07 Bad IOCB size, dopen (Disk)
- 08 Bad IOCB size, dclose (Disk)
- 09 Bad IOCB size, dran (Disk)
- 0a Bad block number, dran (Disk)
- 0b Bad page address, dran (Disk)
- 0c Device not open, dran (Disk)  
Device not open, dseq (Disk)  
Device not open, dphy (Disk)  
Device not open, format (Format)  
Device not open, wphy (DR11W)  
Device not open, aran (Arpanet)  
Device not open, vtec (Versatec)
- 0d Bad IOCB size, dseq (Disk)
- 0e Bad block number, dseq (Disk)
- 0f Bad page address, dseq (Disk)
- 11 Bad IOCB size, dphy (Disk)
- 12 Bad byte count, dphy (Disk)
- 13 Bad fpage address, dphy (Disk)
- 14 Bad block number, dphy (Disk)
- 15 Bad next IOCB pointer, dphy (Disk)
- 16 Bad data chain bit, dphy (Disk)

---

**Anhang 1: I/O-Fehler-Codes**

---

17	Bad page address, dphy (Disk)
18	Nicht belegt
19	Bad device number, tape (Tape)
1a	Bad command number, tape (Tape)
1b	Bad IOCB size, topen (Tape)
1c	Bad IOCB size, tclose (Tape)
1d	Bad IOCB size, tran (Tape)
1e	Bad block number, tran (Tape)
1f	Bad page address, tran (Tape)
20	Device not open, tran (Tape) Device not open, tseq (Tape) Device not open, tphy (Tape)
21	Bad IOCB size, tseq (Tape)
22	Bad block number, tseq (Tape)
23	Bad page address, tseq (Tape)
24	Nicht belegt
25	Bad IOCB size, tphy (Tape)
26	Bad byte count, tphy (Tape)
27	Bad fpage address, tphy (Tape)
28	Bad next IOCB pointer, tphy (Tape)
29	Bad data chain bit, tphy (Tape)
2a	Bad page address, tphy (Tape)
2b	Nicht belegt
2c	Bad command number, ether (Ethernet)
2d	Bad IOCB size, eopen (Ethernet)
2e	Bad physical address, eopen (Ethernet)
2f	Bad IOCB size, eclose (Ethernet)



**Anhang 1: I/O-Fehler-Codes**

30	Bad IOCB size, eran (Ethernet)
31	Bad page address, eran (Ethernet)
32	Device not open, eran (Ethernet)
33	Bad IOCB size, stats
34	Bad page address, stats
35	Bad IOCB size, dtime
36	Bad page address, dtime
37	Bad IOCB size, lgadd
38	Bad page address, lgadd
39	Bad IOCB size, dgadd
3a	Bad page address, dgadd
3b	Bad IOCB size, nltest
3c	Bad IOCB size, cdtest
3d	Bad IOCB size, version
3e	Bad IOCB size, chaninit
3f	Bad ISR field, chaninit
40	Bad sub-channel field, chaninit
41	Bad IOCB size, load
42	Bad password field, load
43	Bad tape number, load
44	Bad disk number, load
45	Bad IOCB size, boot
46	Bad password field, boot
47	Bad device number, boot
48	Pointer not aligned, boot
49	Bad IOCB size, format

---

### Anhang 1: I/O-Fehler-Codes

---

4a	Bad password field, format
4b	Bad cylinder number, format
4c-4f	Nicht belegt
50	Bad IOCB size, uib
51	Bad password field, uib
52	Bad device number, uib
53	Nicht belegt
54	Bad command number, print (Printer)
55	Bad IOCB size, popen (Printer)
56	Bad IOCB size, pclose (Printer)
57	Bad IOCB size, pran (Printer)
58	Bad buffer address, pran (Printer)
59	Bad byte count, pran (Printer)
5a	Device not open, pran (Printer)
5b	Read bit not set, stats
5c	Read bit not set, dtime
5d	Read bit set, lgadd
5e	Read bit set, dgadd
5f	Bad IOCB size, table
60	Bad table number, table
61	Pointer not aligned, table
62	Bad head number, format
63	Bad segment, dopen (Disk)
64	Tape not initialized, tclose (Tape)
65	Bad device number, dr11w (DR11W/Versatec)
66	Bad IOCB size, wopen (DR11W/Versatec)

**Anhang 1: I/O-Fehler-Codes**

- 67 Bad IOCB size, wclose (DR11W/Versatec)
- 68 Bad IOCB size, wphy (DR11W/Versatec)
- 69 Bad byte count, wphy (DR11W/Versatec)
- 6a Bad fpage address, wphy (DR11W/Versatec)
- 6b Bad next IOCB pointer, wphy (DR11W/Versatec)
- 6c Bad data chain bit, wphy (DR11W/Versatec)
- 6d Bad page address, wphy (DR11W/Versatec)
- 6e Bad command number, dr11w (DR11W/Versatec)
- 6f Bad IOCB size, wfs (DR11W/Versatec)
- 70 Bad IOCB size, shift
- 71 Bad password field, shift
- 72 Bad command number, arp (Arpanet)
- 73 Bad IOCB size, aopen (Arpanet)
- 74 Bad page address, aopen (Arpanet)
- 75 Bad IOCB size, aclose (Arpanet)
- 76 Bad IOCB size, aran (Arpanet)
- 77 Bad page address, aran (Arpanet)
- 78 Bad record size field, load
- 79 Bad dchain bit, dchain
- 7a-7e Nicht belegt
- 7f Out of sequence IOCB, stamp

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

---

## Anhang 1: I/O-Fehler-Codes

---

Fehler im Gerätetyp 04 (Bad DCHAIN IOCB) produzieren die folgenden Ursache-Codes:

- 01 Can't read the IOCB, datafill
- 02 Bad next IOCB pointer, dnext
- 03 Bad IOCB size, phy
- 04 Bad next IOCB pointer, phy
- 05 Bad data chain bit, phy
- 06 Bad page address, phy

Fehler im Gerätetyp 06 (Disk Error) produzieren die folgenden Ursache-Codes:

- 10 Disk not ready
- 11 Invalid disk address
- 12 Seek error (actually an address error)
- 13 ECC error, data field
- 14 Invalid command code
- 15 Nicht belegt
- 16 Invalid sector in command
- 17 SPARE
- 18 Bus timeout (greater than 1ms)
- 19 Nicht belegt
- 1a Disk unit is write protected
- 1b Unit not selected (power-off/cable connection)
- 1c No address mark, header field
- 1d Nicht belegt
- 1f-22 Nicht belegt
- 23 Uncorrectable ECC error, data field

**Anhang 1: I/O-Fehler-Codes**

24-25	SPARE
26	No sector pulse
27	Data overrun
28	No index pulse on write format (it has a 65ms timeout)
29	Sector not found
2a	ID field error, wrong head (address error)
2b	Invalid sync in data field
2c	Invalid sync in header field
2d	Seek timeout error (it has a 500ms timeout)
2e	Busy timeout (a 500ms timeout on a dual ported drive)
2f	Not on cylinder (drive must be on cylinder within 3s of selection)
30	RTZ timeout (a 3s timeout occurred after a restore command was issued)
31	Bus late for direct write
32-3f	Nicht belegt
40	Unit not initialized
41	Nicht belegt
42	Gap specification error
43-4a	Nicht belegt
4b	Seek error (actual hardware type)
4c	Mapped header error (no sector pulse found on track to be mapped)
4d-50	Nicht belegt
51	Bytes/sector specification error (bytes/sector in UIB exceeded buffer)
52	Interleave specification error (the interleave factor in the UIB fails a sanity check)

e. „Weitergabe sowie Verweigerung dieser Unterlage, Verwertung und Mitteilung ihres Inhalts nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlung führt zur Suspendierung des Benutzers und die Rechte in dem Fall der Patentierung oder Gebrauchsmustereintragung vorbehalten.“

---

**Anhang 1: I/O-Fehler-Codes**

---

53	Invalid head address
54-5f	Nicht belegt
60	Can't read the UIB
61	Zero error code (when one was expected)
62	Bareturn status
63	Scram error
64	Too many bad cylinders
65	Timeout waiting for a seek interrupt
66	Timeout waiting for a seek complete

Fehler im Gerätetyp 07 (Tape Error) produzieren die folgenden Ursache-Codes:

01	Timed out waiting for Data Busy false
02	Timed out waiting for Data Busy false, Formatter Busy false, and Ready true
03	Timed out waiting for Ready false
04	Timed out waiting for Ready true
05	Timed out waiting for Data Busy true
06	A memory timeout occurred during a system memory reference
07	A blank tape was encountered where data was expected
08	An error occurred in the micro-diagnostic
09	Unexpected EOT on FWD operation or BOT on REV operation
0a	A hard or soft error occurred which could not be eliminated by retry
0b	Read overflow/write underflow in the controller FIFO
0c	Nicht belegt

---

**Anhang 1: I/O-Fehler-Codes**

---

- 0d A read parity error occurred between the drive and the controller
- 0e An error was detected while calculating a checksum on the PROM controller
- 0f A tape timeout occurred...this can occur when attempting to read a larger record than was written or writing to a bad tape
- 10 Tape unit not ready
- 11 No write enable ring on write command
- 12 Nicht belegt
- 13 No diagnostic jumper installed while attempting a diagnostic command
- 14 An attempt was made to link from a command that doesn't allow linking
- 15 An unexpected file mark was encountered
- 16 Parameter error...usual cause is a byte count which is zero or too large
- 17 Nicht belegt
- 18 An unidentifiable hardware error occurred
- 19 A streaming read or write operation was terminated by the Operating System or disk
- 20 Tape head is lost
- 21 Tape timeout
- 22 Read after a previous write
- 23 Zero error status when something was expected
- 24 Extra tape interrupt
- 25 Record size not 2k
- 26 Record size not 2k multiple
- 63 SRAM parity error

### Anhang 1: I/O-Fehler-Codes

---

Fehler im Gerätetyp 08 (Printer Error) produzieren die folgenden Ursache-Codes:

01 Device not ready

Fehler im Gerätetyp 09 (Ethernet) produzieren die folgenden Ursache-Codes:

01 Max retries attempted

21 Ethernet driver timeout

22 Bad ethernet frame size

23 Device not open

63 SRAM parity error

Fehler im Gerätetyp 0A (DR11W) produzieren die folgenden Ursache-Codes:

21 Timeout at reset

23 Device not ready

63 SRAM parity error

Fehler im Gerätetyp 0B (Arpanet) produzieren die folgenden Ursache-Codes:

01 Relay change

02 Last bit not received

03 DMA not done

04 Multi-bus memory timeout

05 Multi-bus memory timeout

06 Can't close relay

22 Bad arpanet frame size

63 SRAM parity error



---

**Anhang 1: I/O-Fehler-Codes**

---

Fehler im Gerätetyp 0C (Versatec) produzieren die folgenden Ursache-Codes:

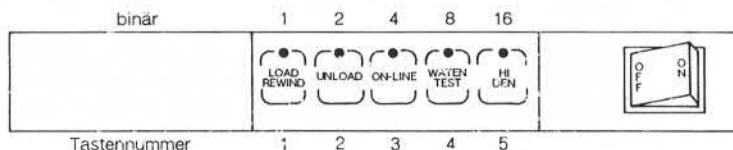
21	Timeout at reset
23	Device not ready
63	SRAM parity error

---



## Anhang 2: Bandlaufwerk: Bedienung/Fehler-Codes

### A2 Bandlaufwerk: Bedienung/Fehler-Codes



#### Automatische Bandenfädung

Gehen Sie bitte in folgenden Schritten vor:

1. Einschalten der Stromzufuhr für das Laufwerk. Nach spätestens acht Sekunden muß das LED der UNLOAD-Sensortaste aufleuchten.
2. Runden Sie den Bandanfang ab – falls nötig mit einer Rändelmaschine.
3. Vergewissern Sie sich, daß der Schreibring – falls benötigt – richtig sitzt.
4. Legen Sie das Band mit dem Schreibring nach unten in das Laufwerk ein und schließen Sie die Klappe. Die Klappe darf die Spule des Bandes nicht berühren.
5. Betätigen Sie die LOAD-Taste. Die Klappe ist nun verschlossen. Nach Beendigung der automatischen Bandenfädung bleibt das LED in der LOAD-Sensortaste konstant erleuchtet.

Sollte einmal die automatische Bandenfädung nicht oder nicht zufriedenstellend funktionieren, unterrichten Sie bitte den für Sie zuständigen Technischen Kundendienst der Nixdorf Computer AG.

## Anhang 2: Bandlaufwerk: Bedienung/Fehler-Codes

---

### Automatisches Rückspulen des Bandes

1. Betätigen Sie die UNLOAD-Taste. Das LED in der ON-LINE-Taste darf nicht leuchten, andernfalls müssen Sie erst diese Taste betätigen.
2. Während des Rückspulens flackert das LED in der UNLOAD-Taste. Die Zugriffsklappe ist verschlossen. Nach Beendigung des Rücklaufs bleibt das LED konstant erleuchtet und die Klappe ist entriegelt.
3. Das Band kann jetzt aus dem Laufwerk entnommen werden.

### Manuelles Rückspulen des Bandes

Versagt der automatische Rückspulmechanismus, gehen Sie bitte folgendermaßen vor:

1. Drehen Sie die Spule entgegen dem Uhrzeigersinn, um das Band auf die Spule zu wickeln.
2. Drücken Sie den Knopf zum Entsperrern und drehen Sie gleichzeitig die Spule entgegen dem Uhrzeigersinn, bis sie sich frei bewegen läßt. Jetzt kann das Band entnommen werden.

**Anhang 2: Bandlaufwerk: Bedienung/Fehler-Codes**

Bedeutung der Leuchtdioden

FEHLER CODE*	ANZEIGE	GRUND
23	Alle LEDs außer TEST blinken.	Der LOAD-Knopf wurde gedrückt, ohne ein Band einzulegen oder die Spule wurde nicht richtig eingelegt.
25	Alle LEDs außer UNLOAD und ON-LINE blinken.	Beim manuellen Einfädeln des Bandes wurde die Spule nicht oft genug gedreht.
26	Alle LEDs außer LOAD und ON-LINE blinken.	Das Einfädeln des Bandes gelingt nicht.
27	Alle LEDs außer ON-LINE blinken.	LOAD oder UNLOAD wurde gedrückt, ohne die Klappe vor dem Laufwerk zu schließen.
28	Alle LEDs außer LOAD und UNLOAD blinken.	Das Laufwerk dreht sich nicht. Entnehmen Sie das Band und legen Sie es neu ein.
29	Alle LEDs außer UNLOAD blinken.	Das Band wurde nicht mit dem Schreibring nach unten eingelegt.
30	Alle LEDs außer LOAD blinken.	Die BOT-Marke (Begin of Tape) wurde innerhalb der üblichen ersten 3,5 m des Bandes nicht gefunden.
31	Alle LEDs blinken.	Das Einfädeln gelingt nicht. Entnehmen Sie das Band und legen es nochmal in das Laufwerk.

\* Der Fehler-Code ist die dezimale Darstellung der binären Anzeige der LEDs.



---

## Anhang 3: Administrator-Kommandos

---

### A3 Administrator-Kommandos

#### **accept, reject – Annahme von Druckaufträgen freigeben/sperrn**

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 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**

**accept, reject**

---

### Anhang 3: 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 Terminalzeit 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 acctmerg 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 acctmerg zusammengeführt und zu Summen-Abrechnungssätzen zusammengefaßt werden (siehe tacct-Format in acct-Format). Mit prtacct (siehe acctsh) werden alle oder ausgewählte Abrechnungssätze formatiert.



---

### Anhang 3: Administrator-Kommandos

---

**A3**

**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
```

© ...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 Gebrauchsmustererlangung vorbehalten.

### Anhang 3: Administrator-Kommandos

---

**DATEIEN:**

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

**SIEHE AUCH:**

Administrator-Kommandos: `acctcms`, `acctcon`, `acctmrg`, `acctprc`,  
`acctsh`, `diskusg`, `fwtmp`, `runacct`  
Dateiformate: `acct`, `utmp`  
Benutzerkommando: `acctcom`  
Systemaufruf: `acct`

---

## Anhang 3: 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 addiert die Sätze für die 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:
  - Kommandonamen  
Namen der ausgeführten Kommandos.
  - Anzahl  
Anzahl der Ausführungen.
  - Summe kcore-Minuten  
Die Anzahl der Speicherkilobytesegmente, die pro Verarbeitung und Minute der Laufzeit genutzt wurden.
  - Summe der CPU-Minuten  
Gesamte Rechenzeit des Kommandos.
  - Reale Summe  
Gesamtzeit, die das Programm zur Beendigung der Verarbeitung benötigte.
  - Durchschnittliche K-Größe  
Durchschnittswert von Summe kcore-Minuten und Anzahl der Kommandoausführungen.
  - Durchschnittliche CPU-Zeit  
Durchschnittswert von Anzahl der Kommandoausführungen und Summe der CPU-Minuten.

---

### Anhang 3: Administrator-Kommandos

---

Hog-Faktor      Messung des Verhältnisses von Systemverfügbarkeit und Systemnutzung.

Die Angaben werden normalerweise nach der Summe der kcore-Minuten 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.

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

Systemaufruf: acct

---

## Anhang 3: Administrator-Kommandos

---

### acctcon1, acctcon2 – Abrechnung von Terminalzeiten (acctcon)

#### SYNTAX:

/usr/lib/acct/acctcon1 [*Optionen*]

/usr/lib/acct/acctcon2

#### BESCHREIBUNG:

##### Acctcon1

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 Rechenzeit, die während des Arbeitsablaufs gebraucht wurde, unterteilt in Hauptnutzungszeit (Prime) und Nebennutzungszeit (Non-Prime). Die Hauptnutzungszeit ist normalerweise Montags – Freitags von 8 – 17 Uhr. Die Nebennutzungszeit umfaßt die restlichen Stunden und Tage sowie die in der Datei /usr/lib/acct/holidays festgesetzten Tage. Diese Daten können den Erfordernissen entsprechend angepaßt werden. Außerdem wird die Startzeit der Session (numerisch) sowie das Startdatum und die Startzeit ausgegeben.

#### 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 Sessionen 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 3: 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 Sessionen, 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 Sessionen verzeichnet werden. (Siehe auch Administrator-Kommando init und Dateiformat utmp.)
- o *Datei* In die angegebene Datei wird eine Gesamtsatz für die Abrechnungsperiode geschrieben, der die Startzeit, die Beendigungszeit, die Anzahl der Neustarts und die Anzahl der Datumswechsel enthält.

#### Acctcon2

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 3: Administrator-Kommandos

---

#### HINWEIS:

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

#### SIEHE AUCH:

Administrator-Kommandos: acct, acctcms, acctmerg, acctprc, acctsh, fwtmp, init, runacct  
Dateiformate: acct, utmp  
Benutzerkommandos: acctcom, init, login  
Systemaufruf: acct

---

### Anhang 3: 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 3: Administrator-Kommandos

---

SIEHE AUCH:

Administrator-Kommandos: acct, acctcms, acctcon, acctprc, acctsh,  
fwtmp, runacct,  
Dateiformate: acct, utmp  
Benutzerkommando: acctcom  
Systemaufruf: acct

## Anhang 3: Administrator-Kommandos

---

### **acctprc1, acctprc2 – Prozeßabrechnung**

SYNTAX:

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

BESCHREIBUNG:

#### **acctprc1**

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**

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
```

---

### Anhang 3: Administrator-Kommandos

---

DATEIEN:

/etc/passwd

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`, `acctmerg`, `acctsh`,  
`cron`, `fwtmp`, `runacct`

Dateiformate: `acct`, `utmp`

Benutzerkommando: `acctcom`

Systemschnittstelle: `acct`

---

### Anhang 3: Administrator-Kommandos

---

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

SYNTAX:

```
/usr/lib/acct/chargefee Login-Name Anzahl  
/usr/lib/acct/ckpacct [Blöcke]  
/usr/lib/acct/dodisk  
/usr/lib/acct/lastlogin  
/usr/lib/acct/monacct [Monat]  
/usr/lib/acct/nulladm Datei  
/usr/lib/acct/prctmp  
/usr/lib/acct/prdaily [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.

**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,

---

### Anhang 3: Administrator-Kommandos

---

A3

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.

**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/ni-  
te/ctmp), die von acctcon1 erstellt wurde, ausgegeben werden.

**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. 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 **prtacct** können Sie jede gewünschte Summen-Abrechnungsdatei (tacct) formatieren und ausgeben.

---

### Anhang 3: Administrator-Kommandos

---

**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/pacct $incr$`  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.

#### DATEIEN:

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

---

### Anhang 3: Administrator-Kommandos

---

SIEHE AUCH:

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

---

## Anhang 3: 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/mtab` (siehe Dateiformat `mtab`), in der sich die Informationen über die eingehängten Dateisysteme (Mounttabelle) befinden. Außerdem lädt **brc** die programmierbaren Mikroprozessoren mit ihren zugehörigen Programmen.

**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 und die Protokollierung der Systemaktivitäten 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 3: Administrator-Kommandos

---

SIEHE AUCH:

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

### Anhang 3: 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:

Systemaufruf: chdir

---

## Anhang 3: 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-Kommando: fsck  
Dateiformat: fs

© Microsofts schriftliche Genehmigung dieser Unterlagen, Vorwertung und Mitführung ihres Inhalts steht gestattet, soweit nicht ausdrücklich zu beschränken. Zuwiderhandlungen verpflichten zu Schadensersatz. Alle Rechte für den Fall der Patenterteilung oder Gebrauchsmustererteilung vorbehalten.

### Anhang 3: Administrator-Kommandos

---

#### **cron – Zeit-Dämon**

##### SYNTAX:

`/etc/cron`

##### BESCHREIBUNG:

Cron führt Kommandos an vorgegebenen Tagen und Uhrzeiten entsprechend den Anweisungen in der Datei `/usr/lib/crontab` aus. 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.

Die Datei `crontab` besteht aus Zeilen zu je sechs Feldern. Die Felder sind durch Leerzeichen oder Tabs voneinander getrennt. Die ersten fünf Felder enthalten Zahlen mit folgender Bedeutung:

- Feld 1: Minute (0-59)
- Feld 2: Stunde (0-23)
- Feld 3: Tag (1-31)
- Feld 4: Monat (1-12)
- Feld 5: Wochentag (0-6, wobei 0=Sonntag ist)

Jedes Feld kann folgenden Inhalt haben:

- Eine Zahl in dem jeweiligen oben angegebenen Bereich.
- Zwei Zahlen, die durch ein Minuszeichen verbunden sind (und einen Bereich von – bis einschließlich angeben).
- Eine Liste von Zahlen, die durch Kommata getrennt sind (womit alle angegebenen Zahlen gemeint sind).
- Einen Stern (der für alle zulässigen Werte steht).

## Anhang 3: Administrator-Kommandos

Das sechste Feld enthält eine Zeichenkette, die von der Shell zum angegebenen Zeitpunkt oder zu den angegebenen Zeitpunkten ausgeführt wird. Ein Prozentzeichen (%) in diesem Feld wird in ein Zeilenende-Zeichen umgesetzt. Nur die erste Zeile (bis zum % oder zum Zeilenende) des Kommandofeldes wird von der Shell ausgeführt. Die weiteren Zeilen werden als Standardeingabe an das Kommando übergeben.

Cron fragt crontab einmal pro Minute ab, um eventuelle Änderungen festzustellen; liegt eine Änderung vor, liest cron diese. Es dauert also jeweils nur eine Minute, bis neue Einträge wirksam werden.

### DATEIEN:

/usr/lib/crontab  
/usr/adm/cronlog

### MELDUNGEN:

Ein Protokoll aller von cron durchgeführten Aktionen wird in /usr/adm/cronlog geführt.

### HINWEIS:

Cron liest crontab nur dann, wenn diese Datei geändert wurde, aber es liest die im Speicher abgelegte Version dieser Datei einmal pro Minute. Ein rationellerer Algorithmus ist denkbar. Der Overhead für cron beträgt ca. ein Prozent der CPU, ausschließlich der von cron ausgeführten Kommandos.

### SIEHE AUCH:

Administrator-Kommando: init

### Anhang 3: Administrator-Kommandos

---

**devnm – Gerätename**

SYNTAX:

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

BESCHREIBUNG:

Devnm dient zur Anzeige der logischen Einheit, welche das Dateisystem mit dem angegebenen Argument enthält.

*Namen* müssen Sie immer als vollen Pfadnamen angeben.

Beispiel:

```
/etc/devnm /u1/doris
```

zeigt an:

```
disk0d /u1/doris
```

wenn /u1/doris der logischen Einheit /dev/disk0d zugeordnet ist.

DATEIEN:

```
/dev/disk*
```

```
/etc/mtab
```

---

## Anhang 3: 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 auf den angegebenen Dateisystemen zur Verfügung stehen. Dazu wertet es die in den Superblöcken geführten Zähler aus. Dateisysteme können entweder mit ihrem Gerätenamen (z. B. /dev/disk0c) oder mit ihrem Wurzelverzeichnisnamen (z. B. /usr) angegeben werden. Geben Sie eine Datei an, z. B. \$HOME, werden die Werte des Dateisystems ausgegeben, in dem diese Datei enthalten ist. Geben Sie kein Argument an, wird der freie Platz in allen eingehängten Dateisystemen ausgegeben.

Optionen:

- i Zusätzliche Ausgabe der Anzahl der freien und belegten I-Knoten.
- l Die Liste der freien Blöcke (free list) wird zusätzlich überprüft, um sicherzustellen, daß die Gesamtzahl der I-Knoten im Superblock korrekt ist.

DATEIEN:

/etc/fstab Liste der eingehängten Dateisysteme

---

## Anhang 3: Administrator-Kommandos

---

### errdead – Fehlersätze aus einem Speicher-Dump extrahieren

SYNTAX:

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

BESCHREIBUNG:

Wenn das System Hardware-Fehler erkennt, 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 zeichnet sie entweder in /usr/adm/errfile auf oder 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 /vmunix angenommen.

Ist *Dumpdatei* angegeben, werden die herausgeholtten Fehlersätze in eine temporäre Datei geschrieben, und errpt wird erzeugt, um eine Auswertung zu erstellen, wobei die temporäre Datei als Eingabe dient. Ist kein Argument angegeben, holt sich errdead die Fehlersätze und eine Panikmeldung des Systems (falls vorhanden) aus dem letzten gezogenen Hauptspeicher-Dump und protokolliert sie in /usr/adm/errfile, so daß diese in einer späteren Auswertung verwendet werden können. Die Protokollierung von Fehlersätzen für spätere Auswertungen kann nur einmal für jeden gezogenen Hauptspeicher-Dump erfolgen.

DATEIEN:

/vmunix	System-Namensliste
/usr/bin/errpt	Auswertungsprogramm
/usr/tmp/errXXX	Temporäre Datei
/usr/adm/errfile	Standard-Fehlerdatei



---

## Anhang 3: Administrator-Kommandos

---

### DIAGNOSE:

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

### SIEHE AUCH:

Administrator-Kommandos: errdemon, errpt

---

### Anhang 3: 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 verloren gehen. `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 3: 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 3: 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 3: 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 `/vmunix` als System-Namensliste verwendet. Nur der Superuser kann `errstop` anwenden.

DATEIEN:

`/vmunix` System-Namensliste

SIEHE AUCH:

Administrator-Kommando: `errdemon`

Benutzerkommando: `ps`

Systemaufrufe: `kill`, `signal`

---

### Anhang 3: 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 3: Administrator-Kommandos

---

- u       Hinter jedem Pfadnamen wird der Login-Namen 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/rrp7
```

### Anhang 3: Administrator-Kommandos

---

#### HINWEISE:

Für einen mehrfach verknüpften I-Knoten wird nur ein einziger der möglichen Pfadnamen generiert, außer wenn die Option `-l` angegeben ist. Ist `-l` 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: `find`, `find`, `ncheck`

Benutzerkommando: `find`



---

### Anhang 3: 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 Bedienumgebung 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 3: 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 aufzuwahren. 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 3: 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 3: Administrator-Kommandos

---

### **format – Online-Plattenformatierer**

#### SYNTAX:

/etc/format

#### BESCHREIBUNG:

Format ist ein Shell-Script, mit dem Platten formatiert werden können. Argumente werden nicht angegeben. Das Kommando fordert alle benötigten Parameter selbst an.

Das zu formatierende Plattenlaufwerk muß bereits im System konfiguriert sein (über COS Frame D, Stub S). Der Plattenformatierer übernimmt alle seine Konfigurationsparameter aus dem UIB. Alle diese Informationen müssen einwandfrei sein, bevor die Platte formatiert werden kann.

Der Formatierer kann im Multi-User-Modus laufen. Allerdings belastet er den IOC so stark, daß er möglichst nur bei ansonsten geringer Systembelastung oder im Single-User-Modus ausgeführt werden sollte.

Je nach Systembelastung und Plattenkapazität kann die Formatierung weit über eine Stunde dauern. Bei Beendigung gibt das Programm die Tabelle der Ersatzspuren aus.

#### DATEIEN:

/dev/rdisk?a	Raw-Disk (Root-Partition)
/etc/iocdfmt	IOC-Plattenformatierungsprogramm
/usr/adm/fmtlog	Kopie des Konsolprotokolls während der Formatierung

#### HINWEIS:

Das Programm prüft nicht, ob das zu formatierende Laufwerk gerade anderweitig benutzt wird.

---

## Anhang 3: 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 3: 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`, `finc`, `volcopy`

Benutzerkommando: `cpio`

---

## Anhang 3: Administrator-Kommandos

---

### fsck – Konsistenzprüfung und interaktive Reparatur von Dateisystemen

#### SYNTAX:

```
/etc/fsck -p [Dateisystem ...]  
/etc/fsck -P [Dateisystem ...]  
/etc/fsck [-b Blocknummer] [-y] [-n] [Dateisystem] ...
```

#### BESCHREIBUNG:

Die erste Form von fsck bereinigt eine Standardmenge von Dateisystemen oder die angegebenen Dateisysteme. Das Kommando wird normalerweise in dem Script /etc/rc beim automatischen Neustart verwendet. In diesem Fall liest fsck die Tabelle /etc/fstab, um die zu prüfenden Dateisysteme ausfindig zu machen. Aufgrund der dort gegebenen Informationen überprüft es Gruppen von Platten parallel und nutzt die überlappende Ein-/Ausgabe soweit wie möglich aus, um die Dateisysteme schnellstmöglich zu prüfen. Normalerweise wird das Wurzel-Dateisystem in Pass 1, andere „Wurzel“-Dateisysteme (Partition a) in Pass 2, andere kleine Dateisysteme in separaten Durchläufen (z. B. die Dateisysteme d in Pass 3 und die Dateisysteme e in Pass 4) und schließlich die größeren Benutzer-Dateisysteme im letzten Pass, z. B. Pass 5, überprüft. Eine Pass-Nummer 0 in fstab bewirkt, daß eine Platte nicht geprüft wird; ebenso werden Partitions, die nicht mit Schreibberechtigung eingehängt sind, nicht überprüft.

Das System sorgt dafür, daß nur bestimmte Arten harmloserer Inkonsistenzen auftreten können, außer wenn Hardware- oder Softwarefehler hinzukommen. Nur folgende Fehler können auftreten:

- Nicht angesprochene I-Knoten,
- Link-Zähler in I-Knoten zu groß,
- Fehlende Blöcke in der Liste freier Blöcke (free list),
- Blöcke in der free list und in Dateien,
- Zähler im Superblock falsch.

---

### Anhang 3: Administrator-Kommandos

---

Dies sind die einzigen Inkonsistenzen, die fsck mit der Option -p korrigiert; wenn das Kommando andere Inkonsistenzen feststellt, endet es mit einem abnormalen Rückgabe-Code; ein automatischer Neustart bleibt dann erfolglos. Für jede korrigierte Inkonsistenz wird mindestens eine Zeile ausgegeben, aus der das Dateisystem, in dem die Korrektur vorgenommen wird, und die Art der Korrektur zu entnehmen sind. Nach einer erfolgreichen Korrektur eines Dateisystems gibt fsck die Anzahl der Dateien in diesem Dateisystem und die Anzahl der belegten und freien Blöcke aus.

Die Option -P hat dieselbe Wirkung wie -p, prüft allerdings keine schreibgeschützten Dateisysteme.

Ohne die Option -p überprüft fsck Inkonsistenzen in Dateisystemen und repariert diese interaktiv. Ist das Dateisystem inkonsistent, wird der Bediener jedesmal um seine Zustimmung gebeten, bevor eine Korrektur versucht wird. Hier sei ausdrücklich darauf hingewiesen, daß bei einer Anzahl von Korrekturmaßnahmen, die mit der Option -p nicht möglich sind, 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 eine Maßnahme entsprechend -n aus.

Fsck führt mehr Konsistenzprüfungen durch als seine Vorgänger check, dcheck, fcheck und icheck zusammen.

Optionen:

-b *Blocknummer*

Der unmittelbar hinter -b angegebene Block gilt als Superblock des Dateisystems. Block 32 ist immer ein alternativer Superblock.

-y Alle Fragen von fsck werden implizit mit ja beantwortet. Diese Option sollten Sie mit äußerster Vorsicht verwenden, da die Verarbeitung auch nach schwersten Fehlern und Verlusten noch fortgesetzt wird.

-n Alle von fsck gestellten Fragen werden implizit mit nein beantwortet; das Dateisystem wird nicht zum Schreiben eröffnet.



### Anhang 3: Administrator-Kommandos

Übergeben Sie fsck beim Aufruf keine Dateisysteme als Parameter, wird eine Standardliste von Dateisystemen aus der Datei /etc/fstab gelesen.

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:  
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 Bereiches.
8. Superblock-Prüfungen:  
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 angesprochen 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 3: Administrator-Kommandos

---

#### DATEIEN:

/etc/fstab Enthält die Standardliste der zu prüfenden Dateisysteme.

#### HINWEISE:

Die l-Knoten-Nummern für . und .. in den einzelnen Verzeichnissen sollten auf Gültigkeit geprüft werden.

Es sollte immer eine Möglichkeit geben, das Kommando fsck -p in jedem Pass zu starten.

#### SIEHE AUCH:

Administrator-Kommandos: newfs, mkfs, reboot

Dateiformate: fstab, fs

## Anhang 3: Administrator-Kommandos

### fuser – Prozesse identifizieren, die eine Datei oder Dateistruktur benutzen

SYNTAX:

*/etc/fuser [-ku] Dateien*

BESCHREIBUNG:

Fuser listet die Prozeß-IDs der Prozesse, die die als Argumente angegebenen Dateien benutzen, auf. Bei blockorientierten Gerätedateien 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, wird außerdem das Signal SIGKILL an jeden Prozeß abgesetzt. Nur der Superuser kann einen Prozeß eines anderen Benutzers terminieren (siehe Systemschnittstelle kill).

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 Standard-Fehlerausgabe geschrieben.

BEISPIELE:

`fuser -ku /dev/dsk1?`

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.

`fuser -u /etc/passwd`

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

### Anhang 3: Administrator-Kommandos

---

**DATEIEN:**

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

**SIEHE AUCH:**

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

---

## Anhang 3: Administrator-Kommandos

---

### **fwtmp, wtmpfix – Bearbeitung von Anschluß-Abrechnungssätzen**

#### SYNTAX:

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

#### BESCHREIBUNG:

##### **fwtmp**

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**

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 3: 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

Benutzerkommando: acctcom, ed

Systemschnittstelle: acct

---

## Anhang 3: Administrator-Kommandos

---

### getty – Terminaltyp, Modi, Geschwindigkeit und Leitungsprozedur einstellen

#### SYNTAX:

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

#### BESCHREIBUNG:

Getty ist ein Programm, das von init aufgerufen wird.

Getty wird von init aufgerufen,

1. um Terminalleitungen zu eröffnen und deren Modi einzustellen,
2. um die Login-Aufforderung auszugeben,
3. um einen Anmeldeprozeß für den Benutzer anzustoßen (der der Login-Aufforderung übergebene Name).

Getty benutzt folgende Dateien:

/etc/gettydefs	Stellt die Terminal-Modi (aus termio entnommen) und die Login-Meldung bereit. Ist die Datei gettydefs nicht vorhanden oder fehlerhaft, verwendet getty bestimmte Standardwerte.
/etc/utmp	Enthält Informationen über angemeldete Benutzer. Utmp gibt auch Auskunft darüber: <ul style="list-style-type: none"> <li>– wann die Maschine gestartet wurde,</li> <li>– wie viele getty-Prozesse darauf warten, Login-Prozesse zu werden.</li> </ul> Beachten Sie, daß /etc/utmp eine bedingte symbolische Verknüpfung auf /etc.attutmp ist.
/etc/wtmp	Enthält kumulative Abrechnungssätze. Init, getty, login und shutdown schreiben Sätze in wtmp.  Beachten Sie, daß /etc/wtmp eine bedingte symbolische Verknüpfung auf /etc.attwtmp ist.

---

### Anhang 3: Administrator-Kommandos

---

Die Vorgehensweise von `getty` finden Sie in den folgenden Abschnitten:

`Getty` 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`, an die `getty` sich selbst anschließen soll. `Getty` benutzt diese Zeichenkette als Name einer Datei im Verzeichnis `/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. Bei Angabe der Option `-l` erwartet `getty` erst das Laden eines Platzprogramms (`DAP4x`), bevor ein Login gestartet werden kann. 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.

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



---

### Anhang 3: Administrator-Kommandos

---

A3

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.

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.

### Anhang 3: Administrator-Kommandos

---

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 Datei in derselben Weise wie im Fall von /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, inittag  
Benutzerkommando: login  
Gerätedatei: tty

---

## Anhang 3: Administrator-Kommandos

---

### halt – Rechnerstop

#### SYNTAX:

/etc/halt [-n] [-q] -y]

#### BESCHREIBUNG:

Nach dem Aufruf von halt werden die noch nicht synchronisierten Daten auf die Platten geschrieben. Anschließend wird der Rechner gestoppt. Ein Reboot des Systems findet nicht statt, selbst wenn im Frame 1 des COS der Autoboot-Schalter auf ON steht.

#### Optionen:

- n Das Synchronisieren wird vor dem Rechnerstillstand **nicht** durchgeführt.
- q Sofortiger Stillstand. Eine ordnungsgemäße Terminierung laufender Prozesse (shutdown) wird nicht vorgenommen.
- y Diese Option müssen Sie angeben, wenn Sie das System von einer anwählbaren Datenstation aus anhalten wollen.

#### SIEHE AUCH:

Administrator-Kommandos: reboot, shutdown

---

## Anhang 3: Administrator-Kommandos

---

### **init, telinit – Initialisierung der Prozeßsteuerung**

SYNTAX:

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

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

BESCHREIBUNG:

#### **init**

Init wird aufgerufen,

- um Scripts zum Hochfahren des Systems auszuführen. Bei diesen Scripts handelt es sich um bcheckrc, brc und rc.
- um das System in den Multi-User-Mode zu bringen. Init wird innerhalb von UNIX als letzter Schritt der Systemstart-Prozeduren aufgerufen.
- um ständig getty-Prozesse zu erzeugen.

Init benutzt die folgenden Dateien:

- /etc/bcheckrc
- /etc/brc
- /etc/rc
- /etc/inittab
- /etc/utmp
- /etc/wtmp

Diese Dateien werden im folgenden beschrieben:

**/etc/bcheckrc** Diese Datei ist eine Prozedur mit Initialisierungskommandos, die einmal vor brc ausgeführt wird. Sie stellt init Programme zur Verfügung zum:

- Vorbereiten des Systems für den Multi-User-Modus;
- Ausführen von fsck.

---

### Anhang 3: Administrator-Kommandos

---

- `/etc/brc` Diese Datei ist ebenfalls ein Script für Initialisierungskommandos, das einmal nach `bcheckrc` und vor `rc` ausgeführt wird. Dieses Script stellt init Prozeduren zur Verfügung zum
- Löschen der Mounttabelle für Dateisysteme;
  - Laden des Mikrocodes in die ITPs.
- `/etc/rc` Diese Datei ist das letzte Script mit Initialisierungskommandos für `init` und wird nach `brc` und `bcheckrc` ausgeführt. `Rc` stellt init Prozeduren zur Verfügung zum
- Einhängen von Dateisystemen;
  - Starten aller Systemdämonen;
  - Starten des Spooler-Schedulers;
  - Starten des Accounting sowie der Protokollierung von Fehlern und Systemaktivitäten;
  - Bereinigen von `tmp`;
  - Eintritt in den Multi-User-Modus.
- `/etc/inittab` Diese Datei stellt init Scripts zur Verfügung zum
- Ausführen von Anlaufaufgaben beim Systemstart;
  - (erneutem) Erzeugen von `getty`.
- Das Kommando `who` greift ebenfalls auf diese Datei zurück, um verschiedene Funktionen unter den 4.2- und System V-Versionen von `init` auszuführen.
- `/etc/utmp` Diese Datei enthält Informationen über angemeldete Benutzer. Ferner gibt sie an,
- wann die Maschine gestartet wurde;
  - wie viele `getty`-Prozesse darauf warten, login-Prozesse zu werden.
- Hinweis: `/etc/utmp` ist eine bedingte symbolische Verknüpfung auf `/etc/attutmp`.

---

### Anhang 3: Administrator-Kommandos

---

`/etc/wtmp` Diese Datei enthält kumulierte Abrechnungssätze. `init`, `getty`, `login` und `shutdown` schreiben Daten in `wtmp`. Hinweis: `/etc/wtmp` ist eine bedingte symbolische Verknüpfung auf `/etc/.attwtmp`.

Wie `init` im einzelnen vorgeht, um einen Prozeß für jeden Terminalanschluß zu kreieren, wird im folgenden beschrieben.

`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, `gettys` 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. 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 SINGLE USER-Modus wird die virtuelle Konsole `/dev/syscon` zum Lesen und Schreiben eröffnet und das Kommando `/bin/su` sofort aufgerufen.

### Anhang 3: Administrator-Kommandos

Zum Verlassen des SINGLE USER-Modus gibt es zwei Möglichkeiten:

- Die Shell wird durch ein Dateiende-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.

Wird init beim Systemstart und immer wenn es vom SINGLE USER-Modus in normale Run-Level umschaltet, setzt es die ioctl-Stati 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 Verarbeitung von inittab erfolgt. So ist es möglich, daß gegebenenfalls erforderliche besondere Initialisierungen des Betriebssystems, z. B. das

---

### Anhang 3: Administrator-Kommandos

---

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 aufgrund eines Dateiende-Zeichens, das entweder explizit eingegeben oder durch eine Leitungsunterbrechung erzeugt wird. Wenn init ein Child-Death-Signal empfängt, 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.

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 fest-



### Anhang 3: Administrator-Kommandos

stellt, verschiedene Bereinigungs- und Protokollierungsfunktionen wahrnehmen. Wichtig ist, daß die Netzausfalleinträge keine Geräte benutzen sollten, die nach einem Netzausfall erst initialisiert werden müssen (z. B. dzb-Leitungen).

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.

#### Die Datenbasis: /etc/inittab

Die Datenbasis von `init` in der Datei `/etc/inittab` besteht aus einer beliebigen Zahl von Einträgen der Form:

*ID:Level:Typ:Prozeß*

- ID* Kennzeichnung von ein bis vier Buchstaben Länge, auf die `init` intern zurückgreift, um Einträge in seiner Prozeßtabelle zu benennen. Die ID wird auch in die dynamische Aufzeichnungsdatei `/etc/utmp` und die Protokolldatei `/etc/wtmp` gestellt. Die ID sollte eindeutig sein.
- Level* Gibt an, in welchen Run-Levels `init` sich mit diesem Eintrag befassen soll. `Level` ist eine Folge von Zeichen bestehend aus [0-6a-c]. Jedesmal wenn der interne `Level` von `init` mit einem in `/etc/inittab` angegebenen `Level` übereinstimmt, ist dieser Eintrag aktiv. Stimmt der interne `Level` von `init` mit keinem der angegebenen `Levels` überein, sorgt `init` dafür, daß der Prozeß nicht läuft. Ist das `Level`-Feld leer, ist dies gleichbedeutend mit der Zeichenkette 0123456.
- Typ* Der `Typ` gibt eine weitere Bedingung an, die für die oder von der Ausführung eines Eintrags benötigt wird.
- `off` Der Eintrag soll nicht ausgeführt werden, auch wenn die `Levels` übereinstimmen.

---

### Anhang 3: Administrator-Kommandos

---

- once** Der Eintrag soll nur dann ausgeführt werden, wenn init in einen Level übergeht. Wenn init durch einen Netzausfall oder die Terminierung eines Kindprozesses aktiviert wurde, wird dieser Eintrag also nicht ausgeführt. Nur wenn durch ein Benutzersignal der Wechsel des internen Status von init zu einem anderen Status angefordert wird und in dem neuen Status dieser Eintrag aktiv sein muß, wird dieser Eintrag aktiviert.
- wait** Wait besitzt sämtliche Merkmale von once und veranlaßt init darüber hinaus zu warten, bis der erzeugte Prozeß terminiert, bevor weitere Einträge aus der Datenbasis gelesen werden. Somit ist es möglich, Initialisierungsmaßnahmen durchzuführen, bevor andere Prozesse, die möglicherweise davon betroffen sind, starten können. In der UNIX-Umgebung wird diese Möglichkeit häufig genutzt, um gemeinsam benutzte Speichersegmente zu initialisieren und Semaphore in den gewünschten Zustand zu bringen.
- respawn** Mit respawn wird festgelegt, daß dieser Eintrag solange laufen soll, wie init in dem Run-Level läuft, der im Level-Feld dieses Eintrags verzeichnet ist. Die meisten von init erzeugten Prozesse fallen in diese Kategorie. Alle getty-Prozesse sind als respawn gekennzeichnet. Immer wenn init die Terminierung eines Prozesses, der als respawn gekennzeichnet war, feststellt, erzeugt es einen neuen Prozeß an seiner Stelle.
- boot** Boot-Einträge zeigen bei der Ausführung dasselbe Verhalten wie once-Einträge. Sie werden nur gestartet, wenn init zum ersten Mal in einen numerischen Run-Level übergeht. Üblicherweise haben boot-Einträge eine leere Level-Zeichenkette; dies bedeutet, daß der boot-Eintrag in jedem Fall ausgeführt wird, gleichgültig in welchem Level init beim ersten Mal übergeht. Enthält die Level-Zeichenkette genauere Angaben, z. B. 01, wird der boot-Eintrag nur dann ausgeführt, wenn init in den Run-Level 0 oder 1 als ersten numerischen Level übergeht.

## Anhang 3: Administrator-Kommandos

### bootwait

Bootwait-Einträge verhalten sich bei der Ausführung genauso wie wait-Einträge, und sie werden ebenso wie boot-Einträge nur dann ausgeführt, wenn init zum ersten Mal in einen numerischen Run-Level übergeht.

### power

Power-Einträge verhalten sich wie once-Einträge und werden aktiviert, wenn init ein SIGPWR-Signal (19) empfängt und sich in einem Status befindet, der mit einem der aktiven Stati des Eintrags übereinstimmt.

### powerwait

Powerwait-Einträge verhalten sich wie wait-Einträge und werden aktiviert, wenn init ein SIGPWR-Signal empfängt und sich in einem Status befindet, der mit einem der aktiven Stati des Eintrags übereinstimmt.

### initdefault

Initdefault weicht insofern vom Üblichen ab, als dieser Eintrag keinen zu erzeugenden Prozeß spezifiziert. Dieser Eintrag gibt nur an, in welchem Level init anfangs übergehen soll, wenn es beim Programmladen hochgefahren wird. Dadurch ist es möglich, das System neu zu starten, ohne daß ein Bediener Eingaben an der Systemkonsole vornehmen muß, wenn dies gewünscht wird. Wenn kein initdefault-Eintrag vorhanden ist, fragt init bei der Systemkonsole /dev/syscon nach dem anfänglichen Run-Level. Neben den numerischen Levels können auch die Single-User-Levels [s und S] angegeben werden.

### Prozeß

Im Prozeßfeld steht die Aktion, die init von einer Shell immer dann verlangt, wenn der Eintrag aktiviert wird. Der Zeichenkette im Prozeßfeld wird das Präfix `exec` vorangestellt, so daß jeder Eintrag anfangs immer nur einen Prozeß erzeugt. Sodann benutzt init die Kommandos `fork` und `exec`.

```
sh -c "exec Prozeß"
```

Dies bedeutet, daß die Prozeß-Zeichenkette die gesamte sh-Syntax voll ausnutzen kann. Die einzigen Besonderheiten ergeben sich aus `exec`, das der Zeichenkette vorangestellt wurde, und außerdem gibt es anfangs keine Standardeingabe,

init, telinit

---

### Anhang 3: Administrator-Kommandos

---

Standardausgabe oder Standardfehlerausgabe. Das Voranstellen von `exec` bedeutet, daß der Benutzer bestimmte Kommandos anders formulieren muß, wenn ein einzelner Eintrag mehrere Prozesse erzeugen soll. Soll bei einem Netzausfall beispielsweise eine Liste der aktuellen Systembenutzer erstellt und diese mit dem Kommando "who | mail root" an Root gesendet werden, so muß dies folgendermaßen angegeben werden:

```
pf::powerwait:sh -c "who | mail root"
```

Würden Sie einfach "who | mail root" eingeben, so würde dies als "exec who | mail root" ausgeführt, und nur der Prozeß who würde erzeugt, bevor die sh terminiert. Das Fehlen von Standardeingabe und Standardausgabe muß durch deren explizite Angabe ausgeglichen werden. Ein Beispiel hierfür ist die Ausführung des `blog`-Programms als `bootwait`-Eintrag, wenn das System hochgefahren wird. Da der Bediener hier Eingaben vornehmen muß, steht es in `/etc/inittab` wie folgt:

```
bl::bootwait:/etc/blog > /dev/syscon > /dev/syscon2 >&1
```

#### Levels

Es gibt folgende Run-Levels: die sieben numerischen Levels 0, 1, 2, 3, 4, 5 und 6, die drei temporären Levels a, b und c sowie den Single-User-Level s und S. Normalerweise läuft `init` in einem numerischen Run-Level. Wie ein bestimmter Level im einzelnen verwendet wird, hängt voll und ganz von der Datenbasis und dem System-Administrator ab. Bei den temporären Levels können bestimmte Einträge auf Anforderung gestartet werden, ohne daß dadurch Prozesse beeinflusst werden, die auf einem bestimmten Level gestartet wurden. Die temporären Levels kehren sofort wieder zu dem vorherigen numerischen Level zurück, nachdem alle Einträge in der Datenbasis daraufhin überprüft wurden, ob sie auf dem temporären Level gestartet werden sollen. Wird ein Eintrag durch Übergang zu einem temporären Level gestartet, wird er unabhängig von jedem späteren Level-Wechsel durch `init` – außer dem Wechsel in den Single-User-Modus. Wollen Sie einen Prozeß, der als `respawn`-Prozeß gestartet wurde, mit `kill` beenden, ohne in den Single-User-Level zu gehen, haben Sie nur die Möglichkeit, die Datenbasis zu ändern und den Eintrag zu deaktivieren.

---

### Anhang 3: Administrator-Kommandos

---

Der Single-User-Level ist der einzige Level, der unabhängig von der Datenbasis ist. Deshalb ist dies auch kein Run-Level im üblichen Sinne. Im Single-User-Modus erzeugt `init` einen `su`-Prozeß auf der Systemkonsole, und dies ist der einzige Prozeß, der im Single-User-Modus weitergeführt wird. Der Single-User-Modus kann in `init` an zwei verschiedenen Stellen aufgerufen werden. Wird er beim Programmieren aufgerufen, so gibt dies dem Bediener die Möglichkeit, die Dateisysteme zu untersuchen, ohne daß `init` zu einer Datei-E/A veranlaßt werden muß, was zu weiteren Problemen führen könnte. `init` versucht nicht, `/etc/utmp` neu anzulegen oder auf `/etc/wtmp` zuzugreifen, bevor es diesen anfänglichen Single-User-Modus verlassen hat. Wird der Single-User-Level zu irgendeinem anderen Zeitpunkt aufgerufen, nimmt `init` die Aufzeichnungen in den Protokollierungsdateien vor.

Der System-Administrator fordert `init` auf, den Level zu wechseln, indem eine Kopie von `init` ausgeführt wird. `/etc/init` wird mit `/etc/telinit` verknüpft, und dies wird gewöhnlich über den `telinit` erreicht. `init` kann nur von `Root` oder einer privilegierten Gruppe ausgeführt werden. Immer wenn `init` zu laufen beginnt und feststellt, daß seine Prozeß-ID nicht 1 ist, sieht es sich als eine vom Benutzer aufgerufene Kopie an, die ein Signal an das echte `init` absetzen muß. Das Kommando hat die Form

```
telinit [0123456sSqQabc]
```

Das aus einem einzelnen Zeichen bestehende Argument gibt das Signal an, daß an `init` abgesetzt werden soll. Wenn aufgrund des Kommandos in den Single-User-Modus `S` oder `s` gewechselt werden soll, verknüpft `init` auch `/dev/syscon` mit dem Terminal, an dem das Kommando abgesetzt wurde, so daß dies zur virtuellen Systemkonsole wird. Dadurch wird gewährleistet, daß weitere Meldungen von `init` dem Terminal zugeleitet werden, an dem der Bediener sich befindet. Bei dieser Neuverknüpfung sendet es ferner eine Meldung an `/dev/systty`, die besagt, daß die Konsole mit einem anderen Terminal verknüpft wird, so daß dieser Umstand an der physischen Konsole belegt ist.

---

## Anhang 3: Administrator-Kommandos

---

### Ereignisse, die init anstoßen

Es gibt vier Ereignisse, die init anstoßen: Booten, Netzausfall, Terminierung eines Kind-Prozesses oder Signal2.

1. Booten  
Init arbeitet im Boot-Status, bis es zum ersten Mal in einen numerischen Status übergeht. Init kann nicht ein zweites Mal in den Boot-Status übergehen. Die Kommandos, die mit dem Level boot und bootwait eingetragen sind, werden ausgeführt, wenn zum ersten Mal in einen numerischen Status gewechselt wird, sofern die Levels übereinstimmen.
2. Netzausfall  
Bei jedem Netzausfall sendet das Betriebssystem ein Signal SIGPWR an alle Prozesse. Init führt dann Kommandos der Typen power und powerfail aus.
3. Terminierung eines Kindprozesses  
Jedesmal wenn ein Kindprozeß von init terminiert, empfängt init ein Signal SIGCLD (18). Bei dem terminierten Kindprozeß gibt es zwei Möglichkeiten: Er war ein direkter Abkömmling von init oder ein Prozeß, dessen eigener Vaterprozeß bereits vorher terminiert hat. Init wird automatisch zum Vater eines Prozesses, wenn der echte Vater des Prozesses vor seinem Kindprozeß terminiert. Init stellt sofort fest, ob der Prozeß einer seiner eigenen Kindprozesse oder ein verwaister (defunct) Prozeß war. War es ein eigener Kindprozeß, nimmt init die notwendigen Aufzeichnungen in seiner internen Prozeßtabelle vor und vermerkt, daß der Prozeß terminiert hat. Ist init beschäftigt, wenn das Signal SIGCLD eintrifft, führt es anschließend die gerade laufende Aktion zu Ende. War init gerade inaktiv, durchsucht es anschließend seine Datenbasis, um festzustellen, ob irgendwelche weiteren Maßnahmen, z. B. das erneute Erzeugen des Prozesses, notwendig sind.
4. Signale  
Init fängt alle Signale auf, die ein Prozeß auffangen kann. Die meisten Signale haben eine spezielle Bedeutung für init und fordern in vielen Fällen irgendeinen Wechsel seines aktuellen Status an. Es gibt ein Signal, das Signal Q, das nur dazu dient, init anzustoßen und zu veranlassen, seine Datenbasis zu untersuchen. Von dieser Möglichkeit wird häufig nach einer Änderung der Datenbasis Gebrauch gemacht, so daß init die neue Änderung sofort wirksam

## Anhang 3: Administrator-Kommandos

macht. Unterläßt man dies, so wird die Änderung erst dann wirksam, wenn init aus einem anderen Grunde angestoßen wird. Abgesehen von der Initialisierungsphase steuert der System-Administrator den internen Level, in dem init läuft, ausschließlich durch Signale.

### Verhalten im normalen Betrieb

Init durchsucht /etc/inittab einmal oder zweimal pro Ereignis, durch das init angestoßen wird. Befindet es sich im Boot-Status oder Powerfail-Status, sucht es die Tabelle einmal nach Einträgen dieser Typen ab, wechselt selbständig in einen normalen Status zurück und sucht erneut.

In einem normalen Status sucht init als erstes /etc/inittab ab und löscht alle Prozesse, die derzeit aktiv sind, aber im aktuellen Level nicht aktiv sein dürfen. Je nachdem, ob init seinen Level wechselt oder nicht, wendet es zur Terminierung seiner Kindprozesse unterschiedliche Methoden an. Wenn init den Level nicht wechselt, erzeugt es per fork einen Kindprozeß für jeden abzubrechenden Kindprozeß und veranlaßt den neuen Kindprozeß, die Signale an den zu terminierenden Prozeß zu senden. Zum Abbrechen eines Prozesses müssen zwei Signale an diesen gesendet werden. Zunächst wird ein Signal SIGTERM (15) gesendet, so daß der Prozeß noch die notwendigen Bereinigungsmaßnahmen treffen kann, um ordnungsgemäß zu terminieren. Nach einer Wartezeit, die in TWARN (Standardwert 20 Sekunden) definiert ist, wird ein Signal SIGKILL (9) gesendet, welches die unbedingte Beendigung des Kindprozesses bewirkt, wenn dieser noch nicht terminiert hat. Das Erzeugen eines Kindprozesses durch fork zum Abbrechen der Prozesse hat den Vorteil, daß der init-Hauptprozeß nicht warten muß, bis alle abzubrechenden Prozesse tatsächlich beendet sind, bevor neue Prozesse erzeugt werden. Der Nachteil ist, daß dann, wenn eine große Anzahl von Prozessen auf diese Weise terminiert werden soll, nicht mehr auszuschließen ist, daß die Prozeßtabelle des Betriebssystems voll wird, so daß der Systemaufruf fork erfolglos verläuft. Dies wiederum würde init stören und auf jeden Fall eine Wartezeit verursachen. Wenn init den Level wechselt, geht es daher grundsätzlich davon aus, daß möglicherweise zahlreiche Prozesse abgebrochen werden müssen und sendet deshalb die Signale selbst, wartet die vorgeschriebenen 20 Sekunden und sendet das abschließende Abbruchsignal, be-

---

### Anhang 3: Administrator-Kommandos

---

vor es seine Arbeit fortsetzt. Sobald die alten Prozesse gelöscht sind, nimmt init einen Eintrag in seinen Aufzeichnungsdateien vor, sofern es den Level wechselt. An dieser Stelle geht es entweder in den Single-User-Modus über oder sucht seine Datenbasis ab, um festzustellen, ob Prozesse im aktuellen Level und im aktuellen Status erzeugt werden müssen. Im normalen Betriebszustand sucht init nach Einträgen der Typen `off`, `once`, `wait` oder `respawn`.

Ist das Absuchen der Datenbasis im normalen Status beendet, ist init bereit, ein anderes Ereignis auszuwerten. Um zu gewährleisten, daß ein Benutzer, der sich soeben abgemeldet hat, seine Dateien auf der Platte aktualisiert hat und daß die verwaltungsmäßigen Aufzeichnungen auf der Platte ebenfalls aktualisiert sind, führt init einen Systemaufruf `sync` aus und macht dann eine Pause, bis es aus irgendeinem Grunde wieder angestoßen wird.

Wird init nach einer Pause angestoßen und aufgefordert, in den Single-User-Modus umzuschalten, sichert es alle `ioctl`-Informationen über die Systemkonsole in der Datei `/etc/ioctl.syscon`, bevor alle weiteren Kindprozesse gelöscht werden. Diese Maßnahmen sind notwendig für den Fall, daß das System heruntergefahren wird, damit in diesem Fall der neue `init`-Prozeß weiß, wie die Systemkonsole eingerichtet werden muß und wie mit ihr kommuniziert wird. Es ist vorteilhaft, wenn die Baudrate und die Terminalspezifikationen nicht geändert werden müssen, wenn man das System remote gestartet wird. Da init den `ioctl`-Status der Systemkonsole auch über einen Neustart des Systems hinweg erhält, sind die beim Neustart ausgegebenen Meldungen für den Bediener lesbar, gleichgültig womit die Systemkonsole gerade verknüpft ist.

Alle verbalen Meldungen von init werden an `/dev/syscon` gesendet. Effektiv sendet init die Meldungen nicht selbst, sondern erzeugt durch `fork` einen Kindprozeß, der die Meldungen absetzt. Dies geschieht, weil init niemals eine Terminalleitung eröffnen darf, da ihm andernfalls ein steuerndes Terminal zugeordnet wird. Da init kein steuerndes Terminal hat, kann es `getty`-Prozesse erzeugen, die anfangs kein steuerndes Terminal haben. Wenn ein solches `getty` sein zugeordnetes Terminal eröffnet, wird dieses Terminal das steuernde Terminal für den `getty`-Prozeß und seine Kindprozesse. Benötigt init in der Initialisierungsphase Eingabedaten vom System-Administrator, eröffnet der Kindprozeß,



---

## Anhang 3: Administrator-Kommandos

---

**A3**

der nach dem Run-Level fragt, /dev/systty, welches immer die physische Systemkonsole ist, bevor die virtuelle Systemkonsole /dev/syscon eröffnet wird. Dadurch ergibt sich, daß /dev/systty das steuernde Terminal des Kindprozesses wird. Wird der Rechner hochgefahren und ist /dev/syscon nicht mit /dev/systty verknüpft oder /dev/syscon nicht arbeitet (z. B. weil eine Datenverbindung beim Neustart ausgefallen ist), kann der Administrator an /dev/systty die Steuerung wieder übernehmen, indem er das Zeichen <DEL> eingibt. Dadurch wird ein Signal SIGINIT (2) an den Kindprozeß gesendet, der /dev/systty wieder mit /dev/syscon verknüpft und erneut nach einem Run-Level fragt, und zwar diesmal an der physischen Systemkonsole.

### Telinit

Mit telinit, das auf /etc/init gelinkt 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 3: Administrator-Kommandos

---

#### DATEIEN:

/etc/inittab

/etc/utmp

/etc/wtmp

/etc/ioctl.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

Systemaufruf: kill

---

## Anhang 3: Administrator-Kommandos

---

### install – Installationskommandos

#### SYNTAX:

```
/etc/install [-c Verzeichnis A] [-f Verzeichnis B] [-r] [-n Ver-
              zeichnis C] [-o] [-s] Datei [Verzeichnis X ...]
```

#### BESCHREIBUNG:

Das Kommando `install` wird meist in Makedateien verwendet, um eine Datei (aktualisierte Zielfeile) an einer bestimmten Stelle innerhalb eines Dateisystems zu installieren. Eine Datei wird installiert, indem man sie in das entsprechende Verzeichnis kopiert und dabei 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 `install` 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` eins oder mehrere Verzeichnisse an, werden diese Verzeichnisse vor den Verzeichnissen der Standardliste durchsucht.

#### Optionen:

##### `-c Verzeichnis A`

Installiert `Datei` in dem 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 3: 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 3: 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`

Systemaufruf: `signal`

### Anhang 3: 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

Systemaufrufe: link, unlink

## Anhang 3: 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 3: 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 3: 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 3: 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 3: 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 angenommenen 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.

© Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Verbreitung inwiefern nicht gestattet, soweit nicht ausdrücklich zugelassen. Nachdruck, Verbreitung, Speicherung, Kopieren, Weitergabe, Verbreitung, Patentierung oder Gebrauchsmustereintragung vorbehalten.

### Anhang 3: Administrator-Kommandos

---

DATEIEN:

/usr/spool/lp/\*

SIEHE AUCH:

Administrator-Kommandos: accept, lpadmin

Benutzerkommandos: enable, lp, lpstat

### Anhang 3: Administrator-Kommandos

A3

#### **makesys – Generierung eines neuen UNIX-Kerns**

**SYNTAX:**

*makesys Benutzerzahl Optionen*

**BESCHREIBUNG:**

Mit makesys kann ein neuer UNIX-Kern konfiguriert und aufgebaut werden. Bei diesem Kommando handelt es sich um eine Zwischenlösung, die in einem späteren Release durch eine flexiblere und allgemeiner anwendbare Funktion ersetzt wird. Die Argumente von makesys sind die maximale Anzahl Benutzer, die sich gleichzeitig im System anmelden können, und eine Liste von Optionen für die Softwarekonfiguration.

Die folgenden Optionen, die in beliebiger Reihenfolge angegeben werden können, sind möglich:

- N Das Netz-Software-Paket (NSP) soll mit konfiguriert werden.
- Q Das quotas-Dateisystem soll mit konfiguriert werden.
- D Das Debugger-Paket für den Betriebssystemkern soll mit konfiguriert werden.

Hinweis: Diese Maßnahme vergrößert den Kern erheblich. Benutzen Sie diese Option nur in Absprache mit Ihrem Nixdorf Kundendiensttechniker.

- M Eine Version des Kerns für Multi-Prozessor-Bibliotheken soll erstellt werden.
- W Der PIP-Driver (DR11-W) soll mit konfiguriert werden.

**VER=Version**

Der Versionsname dieses Kerns. Version erscheint in der utsname-Struktur und im Namen der Ausführungsdatei (siehe unten). Wird diese Option nicht angegeben, wird eine Versionsnummer aus dem Tagesdatum konstruiert, an das sich ein Buchstabe anschließt, der bei jedem Aufruf von makesys erhöht wird, z. B. 0710a, 0710b usw.

© 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.

---

### Anhang 3: Administrator-Kommandos

---

Der Name der von makesys erstellten ausführbaren Datei hat die Form `osxyyy000.VER`, wobei `yyy` die Benutzerzahl, `000` die Verkettung der angegebenen Konfigurationsparameter und `VER` die angegebene Version ist.

Beispiel:

```
makesys 48 VER=test 1 N Q
```

erstellt die Ausgabedatei

```
TOS35-48NQ.test1
```

Werden keine Argumente angegeben und ist eine makesys-Datei vorhanden, werden die Optionen für makesys aus der makesys-Datei entnommen. Dies ist vorteilhaft, wenn alle oder die meisten Kerne mit derselben Konfiguration erstellt werden. Die makesys-Datei kann wie folgt eingerichtet werden:

```
echo 48 N Q > MAKESYS
```

Die E/A-Konfiguration wird durch eine Gruppe von Definitionen (define) gesteuert, die bei der Kompilierung von `conf.c` im aktuellen Verzeichnis verwendet werden. Diese Parameter werden aus der Datei `CONF` (im aktuellen Verzeichnis) gelesen. Diese Parameter sind meist folgende E/A-Konfigurationskonstanten:

- DTWOSWAP Swappen auf Laufwerk 1, Partition b zulassen.
- DLP Drucker-Treiber konfigurieren.
- DTB Grafiktablett-Treiber konfigurieren.
- DBK Berknet-Treiber konfigurieren.
- DXT 5620-Treiber konfigurieren.
- DNPTY=*NN* *NN* (Standardwert 32) Pseudo-Terminals konfigurieren.

---

### Anhang 3: Administrator-Kommandos

---

Die CONF-Datei kann wie folgt eingerichtet werden:

```
echo -DL -DNPTY=50 > CONF
```

Die Option M bewirkt, daß make im Verzeichnis kernel\_m durchgeführt wird, wo alle Bibliotheken das Suffix lib\_m.a besitzen.

HINWEIS:

Die Option M muß vor allen anderen Optionen stehen.

---

## Anhang 3: Administrator-Kommandos

---

### mkfs – Dateisystem aufbauen

SYNTAX:

*/etc/mkfs Gerätedatei Größe [optionale Argumente]*

BESCHREIBUNG:

Mkfs baut in der angegebenen Gerätedatei ein Dateisystem mit einem Root-Verzeichnis (/) und einem lost+found Verzeichnis auf. *Größe* (numerisch) spezifiziert die Anzahl der Plattensektoren, die das Dateisystem belegt. Die Anzahl der benötigten I-Knoten wird selbst berechnet. Ein Bootprogramm in Block 0 wird durch mkfs nicht initialisiert.

Folgende optionalen Argumente können Sie beim Aufruf von mkfs angeben:

- *Sektoren/Spur*  
gibt die Anzahl der Sektoren pro Spur auf der Platte an (Standard = 12).
- *Spuren/Zylinder*  
gibt die Anzahl der Spuren pro Zylinder auf der Platte an (Standard = 20).
- *Blockgröße*  
gibt die Blockgröße der Dateien an. Mögliche Blockgrößen sind 4096 Bytes oder 8192 Bytes (Standard = 8192 Bytes).
- *Fragmentgröße*  
Unterteilung der Blöcke in Fragmente (Standard = 2048 Byte).
- *Zylinder/Gruppe*  
ist die Anzahl der Plattenzylinder pro Zylindergruppe (Standard = 16).
- *Freier Plattenplatz*  
gibt in Prozent das erlaubte Minimum des freien Speicherplatzes an (Standard = 10). Ist dieses Minimum erreicht, darf die Platte nur noch vom Superuser beschrieben werden.
- *Rotationszeit*  
Hat Ihre Platte eine andere Rotationszeit als 60 Umdrehungen/Sekunde, müssen Sie hier die entsprechenden Angaben vorgeben.



---

## Anhang 3: Administrator-Kommandos

---

Beispiel:

Durch die folgenden Kommandos konstruieren Sie die Standard-Dateisysteme auf der TARGON /35 (Plattentyp: Fujitsu Eagle):

```
root: /etc/mkfs /dev/rdisk0a 10080 12 20 8192 2048 16 10
/usr: /etc/mkfs /dev/rdisk0c 25200 12 20 8192 2048 16 10
/u1: /etc/mkfs /dev/rdisk0c 75360 12 20 8192 2048 16 10
/u2: /etc/mkfs /dev/rdisk0c 75360 12 20 8192 2048 16 10
```

SIEHE AUCH:

Administrator-Kommando: fsck  
Dateiformat: fs

### Anhang 3: 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:

Systemaufruf: *mknod*

### Anhang 3: Administrator-Kommandos

A3

#### mount, umount – Dateisystem ein- und aushängen

SYNTAX:

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

BESCHREIBUNG:

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

Soll das Dateisystem nur lesbar sein, so ist die Option *-r* anzugeben.

Mit *umount* können Sie das Dateisystem der logischen Platte *Name* wieder aushängen.

Wenn bei *mount* oder *umount* die Option *-a* angegeben ist, versucht das System, alle in */etc/fstab* beschriebenen Dateisysteme einzuhängen bzw. auszuhängen. In diesem Fall werden *Name* und *Verzeichnis* aus */etc/fstab* entnommen. *Name* in */etc/fstab* ist der Name einer blockorientierten Gerätedatei.

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

Physikalisch schreibgeschützte Dateisysteme und Dateisysteme auf Magnetbändern müssen mit *read-only* eingehängt werden, da andernfalls Fehler auftreten, wenn die Zugriffszeiten aktualisiert werden, gleichgültig ob ein expliziter Schreibvorgang versucht wurde oder nicht.

### Anhang 3: Administrator-Kommandos

---

#### DATEIEN:

/etc/mnttab    Mount-Tabelle (eingehängte Dateisysteme)  
/etc/fstab     Dateisystemtabelle

#### HINWEISE:

Das Mounten von inkonsistenten Dateisystemen führt zum Systemabsturz.

Wird ein Wurzelverzeichnis nicht in ein Verzeichnis eingehängt, werden scheinbar einwandfreie Pfadnamen ungültig.

#### SIEHE AUCH:

Dateiformate: mtab, fstab  
Systemaufruf: mount

---

## Anhang 3: 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 3: 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 3: Administrator-Kommandos

### nlditp – ITP laden

#### SYNTAX:

```
nlditp [-h OxXXXX]
        [-c OxXXXX]
        [-d OxXXXX]
        [-t OxXXXXXXXXX]
        ITP tty [Mikrocode]
```

#### BESCHREIBUNG:

Mit nlditp wird der Mikrocode in den ITP geladen, der ITP in einen laufenden Zustand versetzt und die Zuordnung zwischen dem ITP und der Gruppe von 16 Terminals, die mit der Angabe in *tty* beginnen, vorgenommen. Es wäre nicht richtig, dasselbe *tty*-Argument für verschiedene ITPs zu verwenden. *Mikrocode* ist eine Datei, und zwar gewöhnlich /etc/i.out. Bei fehlender Angabe gilt /etc/i.out als Standardannahme.

Normalerweise wird nlditp nur beim Programmladen in rc-Scripts benutzt. Vor dem ersten Aufruf von nlditp ist es nicht möglich, eins der 16 Terminals, deren Nummern bei *tty* beginnen, zu eröffnen.

Im allgemeinen sind die unten beschriebenen Optionen Bestandteil des ersten Aufrufes in rc. Nlditp kann jedoch erneut aufgerufen werden, um einige der unten beschriebenen Optionen zu ändern. Nach dem ersten Aufruf kann die Zuordnung zwischen den Terminals und dem entsprechenden ITP nicht mehr geändert werden. Ebenso wenig kann der Mikrocode noch einmal geladen werden.

Mit nlditp kann der Träger auf beliebigen Terminals der *tty*-Gruppe bestätigt werden. Die Option -h in Verbindung mit dem nachfolgenden Argument OxXXXX gibt an, bei welchen Kanälen der Träger immer vorhanden sein muß. Um den Träger für den niedrigsten Kanal (es gibt 16 Kanäle) zu bestätigen, muß -h Ox0001 angegeben werden. Mit -h Ox0000 wird der Träger auf allen 16 Kanälen gelöscht. Dieses Vorgehen kann bei Leitungen gewählt werden, bei denen der Träger wegen

---

### Anhang 3: Administrator-Kommandos

---

fehlerhafter Verdrahtung aussetzt oder überhaupt nicht vorhanden ist. Dieses Vorgehen ist besonders nützlich bei Fernwählmodems. Das Bestätigen eines Trägers bei Modems für ankommende Verbindungen ist nicht zu empfehlen, da echte Übergänge im Trägersignal beim Auflegen des Telefons durch den rufenden Partner dann nicht bemerkt werden.

Mit `nlditp` kann die hardwaremäßige CTS-Flußkontrolle für jeden Anschluß permanent aktiviert werden. Um die hardwaremäßige Flußkontrolle für den niedrigsten Anschluß zu aktivieren, würde man `-c Ox0001` angeben. Möchte man die CTS-Flußkontrolle für alle Anschlüsse deaktivieren, so gibt man `-c Ox0000` an. Bei Anschlüssen, die derzeit eröffnet sind, sollte die CTS-Flußkontrolle nicht geändert werden. Als Nebeneffekt der Aktivierung der CTS-Flußkontrolle für einen Anschluß wird das TANDEM-Bit automatisch gesetzt, wenn der Anschluß eröffnet wird. Einem Benutzerprogramm ist es gestattet, das TANDEM-Bit zu löschen. Dann ist es aber sehr wahrscheinlich, daß die Eingabe überläuft und vom System weggeworfen wird.

Mit `nlditp` kann auch festgelegt werden, daß die Signale DTR und RTS für eine gewisse Mindestzeit „low“ bleiben, wenn ein bestimmter Kanal geschlossen wird. Dies ist nützlich für Datenschalter, bei denen die Signale DTR und RTS solange „low“ bleiben müssen, bis der Datenschalter dies erkannt hat. Mit `-d OxXXXX` wird festgelegt, für welche Kanäle dies gilt.

Mit `-t OxXXXXXXXX` wird der zu verwendende Zeitwert festgelegt. Der angegebene Wert stellt die Anzahl der Taktsignale dar. Beispielsweise wird mit `-t Ox0000003c` ein Zeitwert von einer Sekunde definiert (eine 60Hz-Uhr vorausgesetzt). Mit dem Zeitwert 0 kann diese Funktion, die in einem früheren Kommando aktiviert wurde, wieder deaktiviert werden.



---

### Anhang 3: Administrator-Kommandos

---

**A3**

Major-Nummern für ITPs und Ttys sind konfigurationsabhängig. Derzeit gibt es folgende Major-Nummern:

`/dev/itp 1`

`/dev/tty 5`

Die Minor-Nummern der Ttys sind ganze Vielfache von 16, beginnend mit 0. Die höchste Minor-Nummer für die Tty-Gruppe darf nicht größer sein als die im System konfigurierte Grenze, d. h. derzeit 127. Dies bedeutet, daß die Minor-Nummer für Ttys nicht größer als 112 sein kann. Die Minor-Nummer für die ITPs kann die im System konfigurierte Grenze nicht überschreiten, d. h. derzeit 7. Typische ITP-Argumente sind:

`/dev/itp0`

`/dev/itp1`

`/dev/itp2`

`.`

`.`

`.`

`/dev/itp7`

Typische tty-Argumente sind:

`/dev/ttyi00`

`/dev/ttyi16`

`/dev/ttyi32`

`.`

`.`

`.`

`/dev/ttyi112`

---

### Anhang 3: Administrator-Kommandos

---

Zur Konfigurierung der Unterkanäle für die virtuellen Windows brauchen Sie mehr als eine Major-Nummer. Zur Zeit werden zehn Unterkanäle unterstützt. Die niedrigste Major-Nummer ist hier 40. Die folgende Tabelle zeigt Ihnen die Anwendung der Major-Nummern:

TTY	Major	Minor
/dev/tty000	40	0
/dev/tty160	40	160
/dev/tty320	41	64
/dev/tty480	41	224
/dev/tty640	42	128
/dev/tty800	43	32
/dev/tty960	43	192
/dev/tty1120	44	96
/dev/tty1280	45	0
/dev/tty1440	45	160
/dev/tty1600	46	64
/dev/tty1760	46	128
/dev/tty1920	47	128
/dev/tty2080	48	32
/dev/tty2240	48	192
/dev/tty2400	49	96

---

### Anhang 3: Administrator-Kommandos

---

#### MELDUNGEN:

Bei ungültigen Formaten wird eine Fehlermeldung zurückgegeben, die auf das falsche Argument hinweist. Folgende Fehlermeldungen sind möglich:

Cannot stat	Der für das ITP-Gerät angegebene Name ist ungültig.
Not a character device	Das Gerät existiert nicht.
Cannot open	/dev/itp kann nicht eröffnet werden.
Reading kernel state	Das Kopieren ist innerhalb des Kerns fehlgeschlagen.
No such tty	Das angegebene Terminal existiert nicht.
Not a tty	Die angegebene Datei ist keine Terminaldatei.

#### DATEIEN:

/dev/itp  
/dev/tty  
/dev/i.out

#### SIEHE AUCH:

Geräte-datei: itp  
Systemaufruf: probenewitp

### Anhang 3: 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 3: Administrator-Kommandos

---

### reboot – Programm laden

SYNTAX:

```
/etc/reboot [-n] [-q] [-s] [-a]
```

BESCHREIBUNG:

#### Neustart eines laufenden Systems

Wenn UNIX läuft und ein Neustart mit erneutem Programmladen gewünscht wird, wird das System normalerweise mit shutdown heruntergefahren. Arbeiten keine Benutzer am System, kann /etc/reboot verwendet werden. Reboot bewirkt, daß die Platten synchronisiert werden, und anschließend wird ein Multi-User-Neustart (unten beschrieben) eingeleitet. Dabei wird das System geladen, und die Dateisysteme werden automatisch auf Konsistenz geprüft. Wenn alle diese Vorgänge fehlerfrei ablaufen, wird das System schließlich im Multi-User-Modus hochgefahren.

Für reboot stehen folgende Optionen zur Verfügung:

- n Diese Option unterdrückt die Synchronisierung. Sie kann verwendet werden, wenn die Platte oder der Prozessor bekanntermaßen fehlerhaft ist.
- q Lädt das System sofort, ohne laufende Prozesse erst abzuschließen.
- s Lädt das System mit einer Single-User-Shell. Diese Option kann zusammen mit jeder anderen Option verwendet werden.
- a Veranlaßt das Bootstrap-Programm, den Bediener nach dem Gerät und der Datei zu fragen, von denen das System geladen werden soll (Option „askname“).

### Anhang 3: Administrator-Kommandos

---

#### Wiederherstellung nach Netzausfall und Systemabsturz

Normalerweise lädt sich das System nach der Netzeinschaltung und nach einem Systemabsturz selbst, sofern das automatische Programm-laden (auto-boot) von der Konsole her aktiviert ist (Frame 1 von COS). Sodann wird eine automatische Konsistenzprüfung der Dateisysteme durchgeführt, und wenn diese keine Fehler ausweist, geht das System in den Multi-User-Betrieb über.

#### Kaltstarts

Bei Netzeinschaltung werden alle Aufgaben wie das Laden des Mikro-codes, das Initialisieren des Hauptspeichers und das Laden von UIBs in die IOCs für die verschiedenen Platten von COS durchgeführt. COS enthält darüber hinaus mehrere „Softwareschalter“, die verschiedene Aspekte des Programm-ladens steuern: den Auto-Boot-Schalter, das Standard-Gerät von dem gebootet wird und einen Parameter mit Programm-lade-Optionen.

Wenn die Option für das automatische Programm-laden gesetzt ist, wird das System automatisch anhand der Standardwerte für das Boot-Gerät und die Programm-lade-Optionen geladen. Die Maßnahmen beim Programm-laden werden vom Boot-Schalter (siehe unten) gesteuert. Wenn Auto-Boot ausgeschaltet ist, geht COS in Frame 1 über, wo der Benutzer gegebenenfalls das Boot-Gerät, die Programm-lade-Optionen und die Auto-Reboot-Option ändern kann. Durch Eingabe von b (boot), z (run) und <ESC> 0 wird das System gestartet.

Die Werte für den Auto-Boot-Schalter, das Boot-Gerät und die Programm-lade-Optionen können von Frame 1 von COS her gesetzt werden. Diese Werte werden auf der Diskette gespeichert und werden bis zur nächsten Änderung beibehalten. Die Programm-lade-Optionen geben die standardmäßige Programm-lade-Aktion an (kann durch die re-boot-Optionen -s und -a außer Kraft gesetzt werden).

### Anhang 3: Administrator-Kommandos

Optionen:

- 0 Booten bis zum Multi-User-Modus.
- 1 Die Option „askname“ im Bootstrap-Programm wird aktiviert. Es wird eine Single-User-Shell geladen.
- 2 Es wird eine Single-User-Shell geladen.

COS lädt das Bootstrap-Programm (kerninit) von dem angegebenen Boot-Gerät und startet den Prozessor. Das kerninit-Programm sucht, lädt und startet sodann die entsprechende Betriebssystem-Datei. Ist die Boot-Option „askname“ angegeben, fragt kerninit zunächst den Bediener nach dem Gerät und der Datei, von der das Programm geladen werden soll. Das Boot-Gerät und die Datei werden wie folgt spezifiziert:

>>(Kanal,Unterkanal,Einheit,Offset)

Dies hat folgende Bedeutung:

- Kanal* Die xtend-Bus-Adresse des IOC.
- Unterkanal* Der Controller-Typ im IOC (0 ist Platte, 1 ist Band).
- Offset* Der Basis-Offset auf dem Gerät, z. B.:  
(0C,0,0,0)/vmunix

Bei der Angabe des Pfadnamens können die normalen Zeichen verwendet werden.

DATEIEN:

- /vmunix Systemcode
- /boot System-Bootstrap

SIEHE AUCH:

Administrator-Kommandos: crash, fsck, halt, init, shutdown

---

### Anhang 3: Administrator-Kommandos

---

#### runacct – tägliche Abrechnungen ausführen

SYNTAX:

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

BESCHREIBUNG:

Runacct ist die 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 pdaily 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 3: Administrator-Kommandos**

**A3**

PROCESS	Prozeßabrechnungssätze in das tacct.h-Format konvertieren.
MERGE	Die Terminalzeit- 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.

© ..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 3: 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 3: Administrator-Kommandos

---

SIEHE AUCH:

Administrator-Kommandos: acct, acctcms, acctcon, acctmerg, acctprc,  
acctsh, cron, fwtmp  
Dateiformate: acct, utmp  
Benutzerkommandos: acctcom, mail  
Systemaufruf: acct

---

### Anhang 3: 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 [-ubdycwpaqvmA] [-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

```
att/bin/su adm -c "att/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.

Das Shell-Script **sa1** ist eine Variante von `sadc` und dient dazu, die Daten zu erfassen und in der Binärdatei `/usr/adm/sa/sa'tt` 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.

### Anhang 3: Administrator Kommandos

Die Einträge:

```
0 * * * 0,6 att/bin/su adm -c "att/usr/lib/sa/sa1"
0 8-17 * * 1-5 att/bin/su adm -c "att/usr/lib/sa/sa1 1200 3"
0 18-7 * * 1-5 att/bin/su adm -c "att/usr/lib/sa/sa1"
```

in crontab geben während der Arbeitszeit alle 20 Minuten und sonst je-  
de Stunde einmal die Datensätze aus.

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

```
5 18 * * 1-5 att /bin/su adm -c "att/usr/lib/sa/sa2 -s 8:00 -e 18:01 -i  
3600 -A"
```

meldet wichtige Aktivitäten während der Arbeitszeit stündlich.

---

## Anhang 3: Administrator Kommandos

---

Die binäre tägliche Datendatei hat folgende Struktur:

```

struct sa {
    struct sysinfo si;      /* see /usr/include/sys/sysinfo.h */
    int  sz1-node;         /* current entries of 1-node table */
    int  szfile;           /* current entries of file table */
    int  sztext;           /* current entries of text table */
    int  szproc;           /* current entries of proc table */
    int  msz1-node;        /* size of 1-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 [NDEVES] [4]; /* device info for up to NDEVES 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

---

## Anhang 3: Administrator Kommandos

---

### shutdown – System stoppen

SYNTAX:

/etc/shutdown [<n>]

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.

Mit <n> definieren Sie die Anzahl der Sekunden, die vergehen, bevor das System heruntergefahren wird.

Shutdown benutzt die Datei /etc/wtmp, die kumulierte Abrechnungssätze enthält. Init, getty, login und shutdown schreiben Daten in diese Datei.

/etc/wtmp ist eine symbolische Verknüpfung auf /etc.attwtmp.

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 Standard-Meldung ausgegeben.
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.

**shutdown**

### Anhang 3: Administrator Kommandos

---

#### 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 3: Administrator Kommandos

---

### unixboot – Laden eines UNIX-Kerns

#### SYNTAX:

/etc/unixboot

#### BESCHREIBUNG:

Unixboot (auch als kerninit bezeichnet) ist das Bootstrap-Programm, das in den Blöcken 0 bis 7 des Root-Dateisystems enthalten ist und beim Booting vom Console Operating System-Programm (COS) geladen wird. Es hat die Aufgabe, die vmunix-Kerneldatei zu lesen und auszuführen.

In Frame 1 von COS sind zwei Einträge enthalten:

- D Boot-Gerät setzen.  
Dieser Eintrag hat drei Felder: Die IOC XTEND-Bus-Adresse (Standardwert 0x0c), die Unterkanal-Nummer (Platte=0, Band=1; Standardwert=0) und die Laufwerksnummer (Standardwert=0).
- U Programmlade-Optionen setzen.  
In diesem Eintrag werden drei Bits benutzt. Diese Bits können in folgenden Kombinationen auftreten:
  - 0x0 Automatisch in den Multi-User-Betrieb gehen.
  - 0x1 Unixboot stoppen und nach einem Namen fragen.
  - 0x2 Nicht automatisch in den Multi-User-Modus gehen, sondern im Single-User-Modus stoppen.
  - 0x20 Den Kernel-Debugger bei einer Paniksituation bzw. beim Programmieren aufrufen, falls der Kernel mit der Debugger-Option ausgestattet ist.

### Anhang 3: Administrator Kommandos

---

Haben Sie die Programmlade-Option so gesetzt, daß unixboot nach einem Namen fragt, wird eine entsprechende Meldung ausgegeben und auf eine Antwort gewartet. Geben Sie lediglich <CR> ein, wird der Standardkernel geladen (Normalfall). Sie haben folgende Möglichkeiten:

- Angabe eines Boot-Gerätes im angeforderten Format;
- Angabe eines anderen Kernelnamens;
- Angabe eines anderen Boot-Gerätes und eines anderen Kernelnamens.

Ist der Kernelname nicht vorhanden, gibt unixboot eine Liste der Dateien im Verzeichnis aus. Durch Eingabe eines ungültigen Kernelnamens (z. B. xxx) können Sie sich eine Liste der gültigen Kernelnamen ausgeben lassen.

DATEIEN:

/etc/unixboot    Boot-Programm

### Anhang 3: 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:**

- d*Verzeichnis***      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.
- n*Stunden***          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.

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

---

### Anhang 3: Administrator Kommandos

---

- sSystem* Es werden nur Dateien untersucht, die für das System bestimmt sind. Die Option *-s* kann bis zu zehnmal 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.

Uuclean wird im typischen Fall über cron gestartet.

#### DATEIEN:

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

#### SIEHE AUCH:

Administrator-Kommando: cron  
Benutzerkommandos: uucp, uux

### Anhang 3: 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:

- aSystem    System wird in das Teilnetz (Liste der zu überwachenden Systeme) aufgenommen.
- dSystem    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.
- uStunden   Die Verkehrsstatistiken für die angegebene Anzahl Stunden erfassen.
- cSystem    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 Anruferinrichtung), #login ist die Anzahl der erfolglosen Verbindungsversuche wegen negativ verlaufener Anmeldung, #nack ist

**uusub**

---

### Anhang 3: Administrator Kommandos

---

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 uu-sub-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 3: Administrator Kommandos

---

### volcopy, labelit – Dateisysteme unter Kennsatzprüfung kopieren

#### SYNTAX:

```
/etc/volcopy [Optionen] Dateisystem Gerät1 Volname1 Ge-
                rät2 Volname2
```

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

#### BESCHREIBUNG:

##### volcopy

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 3: Administrator Kommandos

---

*Gerät* ist physikalische Plattenabschnitt bzw. das Band (z. B. /dev/rdisk15, /dev/rmt0 usw).

*Volname* ist der physikalische Datenträgername (z. B. pk3, t0122 usw.) 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 fsname[6], volname[6];).

#### labelit

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 3: 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 muß vom Superuser aufgerufen werden, um einen eventuell gesetzten Schreibschutz der Terminals zu brechen.

**DATEIEN:**

```
/dev/tty*
```

**SIEHE AUCH:**

Benutzerkommandos: mesg, write

### Anhang 3: 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 4: Gerätedateien

---

### A4      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 4: Gerätedateien

---

### **init – Kanal zu Prozeß 1**

#### BESCHREIBUNG:

Init ist eine spezielle Gerätedatei, mit deren Hilfe Nachrichten an Prozeß 1 übergeben werden. Sie wird hauptsächlich von Login-Dämonprozessen benutzt, um eine remote Prozeßabrechnung für das Login-Accounting bei System V durchzuführen. Die Übergabe von Nachrichten erfolgt über die Systemschnittstelle ioctl.

#### DATEIEN:

/dev/init

/usr/include/sys/devinit.h

## Anhang 4: Gerädateien

### itp – Intelligenter Terminal-Prozessor

#### BESCHREIBUNG:

Mit Hilfe der Gerädatei /dev/itp werden intelligente Terminalprozessoren (ITPs) mit Mikrocode geladen, getestet und gestartet. Diese Schnittstelle unterstützt die folgenden Systemaufrufe: open, close, read, write und ioctl. Der System Call open ist exklusiv, so daß es jeweils immer nur einen open für eine Datei /dev/itp geben kann.

Alle Lese- und Schreiboperationen auf dem Instruktions- und Datenspeicher des ITP sind Wortoperationen (4 Bytes). Die Adressen müssen an Wortgrenzen ausgerichtet werden, und die Byteanzahl muß ein ganzes Vielfaches von 4 sein.

Die Suchadressen 0 – NUMITPI/4 beziehen sich auf den Instruktionsspeicher des ITP. Die Adressen NUMITPI/4 – NUMITPD/4 beziehen sich auf den Datenspeicher des ITP.

```
#include <sys/itpdefs.h>
```

```
struct ioctlitp {
    int status;
    int value;
    int addr;
}
```

```
ioctlitp;
```

```
ioctl (itpfd, cmd, &ioctlitp)
```

Jedes Kommando führt die gewünschte Funktion aus und aktualisiert ioctlitp.status mit dem Inhalt des Hardware-Statusregisters des ITP (nach Ausführung des Kommandos). ITPUPDATE liest lediglich das Hardware-Statusregister des ITP.

Im allgemeinen liefern fehlgeschlagene Hardware-Port-Operationen den Wert -1 zurück und setzen errno auf EIO oder ENXIO. ENXIO bedeutet, daß die anfängliche Port-Operation zum Lesen des Status fehlgeschlagen ist. Dies bedeutet in der Regel, daß der angegebene ITP nicht an den XtendBus angeschlossen ist oder daß er nicht einwandfrei funktioniert. EIO bedeutet, daß die anfängliche Port-Operation zum Lesen des Status erfolgreich verlaufen ist, die angeforderte Operation aber fehlgeschlagen ist.

---

## Anhang 4: Gerätedateien

---

Kommando:	Beschreibung:
ITPSTEP	Einzelschritte ausführen. Der ITP muß vor Ausführung des Kommandos angehalten werden.
ITPDBGSTEP	Debug-Einzelschritte ausführen. Dieses Kommando führt die Instruktion in <code>ioctlitp.value</code> aus. Der ITP muß angehalten werden, bevor ein Debug-Einzelschritt ausgeführt wird, und danach wird er explizit angehalten.
ITPUPDATE	Status des ITP aktualisieren. ITPUPDATE verläuft nur dann erfolglos, wenn der ITP nicht an den XtendBus angeschlossen ist oder wenn schwerwiegende Hardwareprobleme vorliegen.
ITPHALT	Den ITP anhalten.
ITPRUN	Den ITP starten. Der ITP muß angehalten sein, damit dieses Kommando ausgeführt werden kann.
ITPRESET	Reset der ITP-Hardware.  Achtung: Ein Reset eines laufenden ITPs sollte im Multi-User-Betrieb nicht vorgenommen werden. Geschieht dies doch, werden seine Terminals außer Betrieb gesetzt. Das System muß dann neu gebootet werden.
ITPREAD	Die Instruktions- oder Datenspeicheradresse des ITP, die in <code>ioctlitp.addr</code> angegeben ist, wird gelesen und das Ergebnis in <code>ioctlitp.value</code> abgestellt. Der ITP muß angehalten werden, wenn der Instruktionsspeicher (Adressen 0 bis <code>NUMITPI/4</code> ) gelesen oder in diesen geschrieben werden soll. Dies ist beim Datenspeicher des ITP nicht notwendig. (Diese Schnittstelle sollte nur von den Fachleuten der Nixdorf Computer AG benutzt werden.)
ITPWRITE	Die Daten in <code>ioctlitp.value</code> auf die von <code>ioctlitp.addr</code> angegebene Adresse im Instruktions- oder Datenspeicher des ITP schreiben. Wenn sich die Adresse im Instruktionsspeicher befindet, muß der ITP angehalten werden.

---

## Anhang 4: Gerätedateien

---

**ITPINIT**      Einen laufenden ITP an den Kernel anschließen. Der ITP muß laufen, bevor dieses Kommando ausgeführt werden kann. Dieses Kommando kann nur einmal je ITP abgesetzt werden.

### Trägersignal

Mit den folgenden Kommandos wird das Trägersignal immer für ausgewählte Kanäle geltend gemacht. Nur die unteren 16 Bits sind signifikant.

Beispiel:

```
int mask;  
ioctl (fd, ITPSETCARR, &mask)
```

Kommando:      Beschreibung:

**ITPSETCARR**      Die in der Maske enthaltenen Bits in Oder-Bedingung mit der aktuellen festverdrahteten Trägersignal-Maske setzen. Gegebenenfalls hinzugefügte Kanäle, die derzeit beim Trägersignal blockieren, werden so angestoßen, als ob das Trägersignal soeben aufgetreten wäre.

**ITPCLEARCARR**      Die in der Maske enthaltenen Bits in der festverdrahteten Trägersignal-Maske löschen. Dadurch wird der Zustand des Trägersignals auf eröffneten Kanälen nicht verändert.

**ITPGETCARR**      Die Maske auf die aktuelle festverdrahtete Trägersignal-Maske setzen.

---

## Anhang 4: Gerätedateien

---

### Hardwaremäßige Flußkontrolle

Mit den folgenden Kommandos wird die hardwaremäßige Flußkontrolle (CTS,M2) für ausgewählte Kanäle aktiviert und deaktiviert. Wenn ein Kanal mit hardwaremäßiger Flußkontrolle arbeitet, die über diese Maske aktiviert wird, wird die hardwaremäßige Flußkontrolle beim ersten open für diesen Port eingeschaltet. Dieser Effekt ist ähnlich wie bei TIOCHCTS, allerdings geschieht dies nach dem ersten open, bevor ein ioctl ausgeführt werden kann. Ferner wird das TANDEM-Bit automatisch gesetzt. Das TANDEM-Bit muß gesetzt sein, wenn die Terminal-Eingaberoutinen irgendeine Form der Flußkontrolle ausüben sollen. Nur die unteren 16 Bits sind signifikant.

Beispiel:

```
int mask;  
    ioctl (fc, ITPSETHCTS, &mask)
```

Kommando:      Beschreibung:

ITPSETHCTS      Die in der Maske enthaltenen Bits mit der aktuellen hardwaremäßigen Flußkontrollmaske in Oder-Verbindung setzen.

ITPSETHCTS      Die in der Maske enthaltenen Bits in der aktuellen hardwaremäßigen Flußkontrollmaske löschen.

ITPGETHCTS      Die Maske auf die aktuelle hardwaremäßige Flußkontrollmaske setzen.



## Anhang 4: Gerädateien

### Leitungsunterbrechung

Die folgenden Kommandos beziehen sich auf die Maske für das zwingend herbeigeführte hup (hang up = Trennen der Leitung). Mit der Maske für das zwingend herbeigeführte hup wird gewährleistet, daß die Modemsignale S1 und S2 immer dann für eine festgelegte Zeitspanne ausgeschaltet bleiben, wenn der Kanal geschlossen oder die Baudrate auf 0 gesetzt wird. Nur die unteren 16 Bits sind signifikant.

Beispiel:

```
int mask;
    ioctl (fd, ITPSETHUPMASK, &mask)
```

Kommando:            Beschreibung:

**ITPSETHUPMASK**    Die in der Maske enthaltenen Bits mit der aktuellen Maske für das zwingend herbeigeführte hup in Oder-Verbindung setzen. Die der Maske hinzugefügten Bits werden erst beim nächsten close für den Kanal bzw. dann, wenn die Baudrate auf dem Kanal auf 0 gesetzt wird, wirksam.

**CLEARHUPMASK**    In dieser Maske die Bits löschen, die in der aktuellen Maske für das zwingend herbeigeführte hup gesetzt sind. Bei open blockierende Kanäle, deren Bit soeben gelöscht wurde, beenden die Blockierung und setzen den open fort.

**GETHUPMASK**        Die Maske auf die aktuelle Maske für das zwingend herbeigeführte hup setzen.

Die folgenden Kommandos setzen bzw. holen die minimale Zeit zwischen close und open, damit die Modemstrecke nachkommt.

Beispiel:

```
int ticks;
    ioctl (fd, ITPSETHUPTIME; &ticks)
```

---

## Anhang 4: Gerätedateien

---

### Test der Anbindung des ITP an den Kernel

Kommando:	Beschreibung:
ITPGETDEV	Den Wert auf die Major- und die Minor-Nummer der Terminal-Gruppe des ITP setzen. Das Kommando holt den Wert, der bei dem ITP gesetzt ist.
GETITPSTATE	Den Wert auf den aktuellen Status des ITP entsprechend dem Kernel setzen. Die folgenden Werte werden zurückgegeben: <ul style="list-style-type: none"> <li>-0 Ursprünglicher Zustand. Für diesen ITP ist nie ein ITPINIT ausgeführt worden.</li> <li>-1 Dieser ITP wird gerade initialisiert.</li> <li>-2 Dieser ITP läuft und ein ITPINIT ist erfolgreich ausgeführt worden.</li> <li>-3 Dieser ITP ist abgestürzt oder seine Initialisierung ist fehlgeschlagen.</li> </ul>

### FEHLERMELDUNGEN:

Es folgen die möglichen Werte, auf die errno gesetzt sein kann, wenn ein Fehler auftritt. Die wahrscheinlichen Ursachen der Fehler werden ebenfalls angegeben.

Errno:	Ursache:
[EIO]	Eine Port-Operation ist erfolglos verlaufen, als ein ioctl-Kommando abgesetzt wurde. Der ITP konnte nicht mit ITPHALT angehalten werden. Der ITP konnte nicht mit ITPRUN gestartet werden.
[EINVAL]	Der ITP war nicht angehalten, als ein ITPSTEP angefordert wurde. Der ITP war nicht angehalten, als ein ITPDBGSTEP angefordert wurde. Der ITP lief bereits, als ein ITPRUN angefordert wurde. Beim Anfordern eines ITPSETHUPTIME wurden weniger als 0 Taktsignale angegeben.
[ENXIO]	Der Status des ITP konnte nicht gelesen werden, als ein ITPUPDATE-Kommando abgesetzt wurde.

---

## Anhang 4: Gerätedateien

---

DATEIEN:

/dev/itp

SIEHE AUCH:

Administrator-Kommando: nlditp

Dateiformat: tty

## Anhang 4: 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 virtueller Speicher im Kernel statt physikalischem Speicher angesprochen wird. Der **ps**-Befehl z. B. liest Prozeß- und Benutzerstrukturen aus **/dev/kmem**.

#### DATEIEN:

**/dev/mem**

**/dev/kmem**

---

## Anhang 4: 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 und EOF zurückgemeldet.

#### DATEIEN:

/dev/null

## Anhang 4: Gerätedateien

---

### **termio – allgemeine Terminalschnittstelle**

#### BESCHREIBUNG:

Alle asynchronen Kommunikationsanschlüsse arbeiten mit derselben 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 4: 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 keine anderen Vorkehrungen getroffen, wird er nicht nur beendet, sondern es wird auch eine Core-Image-Datei (mit dem Namen core) im aktuellen Verzeichnis angelegt.  |

---

## Anhang 4: Gerädateien

---

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, 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 4: Gerätedateien

Fällt das Trägersignal (M5) des Modems 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 */
};
```

---

## Anhang 4: Gerädateien

---

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

0	INTR	DEL
1	QUIT	FS
2	ERASE	#
3	KILL	@
4	EOF	EOT
5	EOL	NUL
6	reserved	
7	reserved	

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 4: 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 4: 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
NLDLY	0000400	Zeitverzögerungen für Newline festlegen:
NLO	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	

Ist OPOST gesetzt, werden die Ausgabezeichen so nachverarbeitet, wie dies von den weiteren Flags angegeben wird; andernfalls werden die Zeichen unverändert übergeben.

## Anhang 4: Gerätedateien

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.

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.

## Anhang 4: Gerätedateien

---

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. Im DAP4x-Modus gibt es keine Zeitverzögerung.

Der anfängliche Wert für die Ausgabesteuerung ist „alle Bits gelöscht“.

**Anhang 4: Gerätedateien**

Das Feld `c_cflag` beschreibt die hardwaremäßige Steuerung des Terminals:

CBAUD	0000017	Baudrate:
B0	0	Hangup
B50	0000001	50 Baud
B75	0000002	75 Baud
B110	0000003	110 Baud
B134	0000004	134,5 Baud
B150	0000005	150 Baud
B200	0000006	200 Baud
B300	0000007	300 Baud
B600	0000010	600 Baud
B1200	0000011	1200 Baud
B1800	0000012	1800 Baud
B2400	0000013	2400 Baud
B4800	0000014	4800 Baud
B9600	0000015	9600 Baud
EXTA	0000016	External A
EXTB	0000017	External B
CSIZE	0000060	Zeichenlänge:
CS5	0	5 Bits
CS6	0000020	6 Bits
CS7	0000040	7 Bits
CS8	0000060	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

Die Baudraten 50, 200 und EXTB werden vom ITP nicht unterstützt. Setzt man ein ITP-Terminal auf einen dieser Werte, so wird bei der E/A-Steuerung ein Fehler gemeldet.

---

## Anhang 4: Gerätedateien

---

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.

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.

Der anfängliche hardwaremäßige Wert nach dem Eröffnen ist B300, CS8, CREAD, HUPCL.



## Anhang 4: Gerätedateien

Das Feld `c_iflag` der Argumentstruktur wird von der Leitungsprozedur zur Steuerung der Terminalfunktionen verwendet. Das grundlegende Leitungsprotokoll (0) sieht folgendes vor:

ISIG	000001	Signale aktivieren
ICANON	000002	Verarbeitung von <code>erase</code> und <code>kill</code>
XCASE	000004	Darstellung von Groß-/Kleinbuchstaben
ECHO	000010	Echo aktivieren
ECHOE	000020	Zeilenlöschzeichen als BS-SP-BS zurückmelden
ECHOK	000040	NL nach Zeilenlöschzeichen zurückmelden
ECHONL	000100	NL zurückmelden
NOFLSH	000200	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 (Raw-Mode). 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 4: 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 4: 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 (Dateibeschreibungsnummer, 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 (Dateibeschreibungsnummer, Kommando, Argument)
int arg;
```

## Anhang 4: 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  
Systemaufruf: ioctl

---

**Anhang 4: 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 5: Dateiformate

### A5 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 compt; /* "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 */
    compt ac_untime;  /* acctng user time in clock ticks */
    compt ac_stime;   /* acctng system time in clock ticks */
    compt ac_etime;   /* acctng elapsed time in clock ticks */
    compt ac_mem;     /* memory usage in clicks */
    compt ac_io;      /* chars trnsfrd by read/write */
    compt 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 */
```

---

## Anhang 5: 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 `ac_mem/(ac_stime+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 */
};
```

HINWEIS:

Der `ac_mem`-Wert für ein kurzlebige 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.



---

## Anhang 5: Dateiformate

---

SIEHE AUCH:

Administrator-Kommando: acct

Benutzerkommando: acctcom

Systemaufrufe: acct, exec, fork

---

## Anhang 5: 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 Arrays 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 beim Systemaufruf `stat` erläutert. Die Länge des durch ein Nullzeichen beendeten Pfadnamens `h_name` einschließlich des Nullzeichens wird von `h_namesize` angegeben.

---

## Anhang 5: 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  
Systemaufruf: stat

---

## Anhang 5: 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 */
};
```

Folgende Satzarten sind zulässig:

```
#define E_GOTS      010      /* start for UNIX/TS */
#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_PANIC     021      /* system panic message */
#define E_TERM      077      /* log file termination */
```

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 5: Dateiformate

```

struct estart {
    struct utsname   e_name;      /* system names */
    int              e_memsize;   /* size of physical memory */
    short           e_bconf;     /* block dev configuration */
};

#define eend errhdr /* record header */

struct etimchg {
    time_t          e_ntime;     /* new time */
    int              e_nminwest; /* new minutes west of Greenwich */
    int              e_ndst;     /* new type of dst correction */
    int              e_ominwest; /* old minutes west of Greenwich */
    int              e_odst;     /* old type of dst correction */
};
    
```

Fehlersätze für blockorientierte Geräte haben das folgende Format:

```

struct eblock {
    dev_t           e_dev;       /* logical device number */
    int              e_iodev;    /* physical device number */
    int              e_ops;      /* number read/writes */
    int              e_misc;     /* number other operations */
    ushort          e_unlog;     /* number unlogged errors */
    short           e_bflags;    /* read/write, error, etc */
    daddr_t         e_blkno;     /* logical block number */
    ushort          e_bytes;     /* number bytes to transfer */
    ushort          e_abbytes;   /* number bytes transferred */
    paddr_t         e_memaddr;   /* buffer address */
    ushort          e_devtype;   /* device type */
    ushort          e_rtry;     /* number retries */
    int              e_seek;     /* cylinder/track/sector */
    int              e_status;   /* interrupt status */
};
    
```

Die folgenden Werte werden im e\_bflags-Wort verwendet:

```

#define E_WRITE    0           /* write operation */
#define E_READ    1           /* read operation */
#define E_OTHER   02          /* no I/O pending */
#define E_PHYS    04          /* physical I/O */
#define E_UNREC   0200        /* I/O failed */
    
```

## Anhang 5: Dateiformate

---

Die Gerätetypen, auf denen die Fehler aufgetreten sind, werden wie folgt bezeichnet:

```
#define IOCDISK    0
#define IOCTAPE   1
```

Systempanikmeldungen werden nur von `errdead` protokolliert, wenn im Speicherauszug ein Panik-String gefunden wird. Der Satz besteht aus einem Header, an den sich eine mit Nullzeichen abgeschlossene Zeichenkette anschließt.

Ein Dateibeendigungssatz mit folgendem Format wird hinter dem letzten Fehlersatz in der Datei geschrieben. Hinter dem Beendigungssatz sind folgende Daten ungültig.

```
define eterm errhdr /*record header */
```

SIEHE AUCH:

Administrator-Kommando: `errdemon`

## Anhang 5: Dateiformate

### fs, inode – Format eines Datenträgers für Dateisysteme

#### SYNTAX:

```
#include <sys/types.h>
#include <sys/fs.h>
#include <sys/inode.h>
```

#### BESCHREIBUNG:

Alle Datenträger für Dateisysteme (Platten, 9-Spur-Bänder usw.) besitzen für bestimmte wichtige Informationen ein gemeinsames Format. Jeder dieser Datenträger ist in eine bestimmte Anzahl Blöcke unterteilt. Die Blocklänge ist ein Parameter des Dateisystems. In den Sektoren 0 bis 7 eines Dateisystems liegen primäre und sekundäre Bootstrap-Programme.

Das eigentliche Dateisystem beginnt in Sektor 8 mit dem Superblock. Der Superblock hat das von der include-Datei <sys/fs.h> definierte folgende Format:

```
#define FS_MAGIC 0x011954
struct fs {
    struct fs *fs_link; /* linked list of file systems */
    struct fs *fs_rlink; /* used for incore super blocks */
    daddr_t fs_sbknno; /* addr of super block in filesystems */
    daddr_t fs_cblkno; /* offset of cyl-block in filesystems */
    daddr_t fs_ibknno; /* offset of inode-blocks in filesystems */
    daddr_t fs_dblkno; /* offset of first data after cg */
    long fs_cgoffset; /* cylinder group offset in cylinder */
    long fs_cgmask; /* used to calc mod fs_ntrak */
    time_t fs_time; /* last time written */
    long fs_size; /* number of blocks in fs */
    long fs_dsize; /* number of data blocks in fs */
    long fs_ncg; /* number of cylinder groups */
    long fs_bsize; /* size of basic blocks in fs */
    long fs_fsize; /* size of frag blocks in fs */
    long fs_frag; /* number of frags in a block in fs */
};
```

## Anhang 5: Dateiformate

```

/* these are configuration parameters */
long    fs_minfree;    /* minimum percentage of free blocks */
long    fs_rotdelay;  /* num of ms for optimal next block */
long    fs_rps;       /* disk revolutions per second */
/* these fields can be computed from the others */
long    fs_bmask;     /* "blkoff" calc of blk offsets */
long    fs_fmask;     /* "fragoff" calc of frag offsets */
long    fs_bshift;    /* "lblkno" calc of logical blkno */
long    fs_fshift;    /* "numfrags" calc number of frags */
/* these are configuration parameters */
long    fs_maxcontig; /* max number of contiguous blks */
long    fs_maxbpg;    /* max number of blks per cyl group */
/* these fields can be computed from the others */
long    fs_fragshift; /* block to frag shift */
long    fs_fsbtodb;   /* fsbtodb and dbtofsb shift constant */
long    fs_sbssize;   /* actual size of super block */
long    fs_csmask;    /* csum block offset */
long    fs_csshift;   /* csum block number */
long    fs_nindir;    /* value of NINDIR */
long    fs_inopb;     /* value of INOPB */
long    fs_nspf;      /* value of NSPF */
long    fs_sparecon[6]; /* reserved for future constants */
/* size determined by number of cylinder groups and their sizes */
daddr_t fs_csaddr;    /* blk addr of cyl grp summary area */
long    fs_cssize;    /* size of cyl grp summary area */
long    fs_cgsize;    /* cylinder group size */
/* these fields should be derived from the hardware */
long    fs_ntrak;     /* tracks per cylinder */
long    fs_nsect;     /* sectors per track */
long    fs_spc;       /* sectors per cylinder */
/* this comes from the disk driver partitioning */
long    fs_ncyl;      /* cylinders in file system */
/* these fields can be computed from the others */
long    fs_cpg;       /* cylinders per group */
long    fs_ipg;       /* inodes per group */
long    fs_fpg;       /* blocks per group *fs_frag */
/* this data must be re-computed after crashes */
struct  csum fs_cstotal; /* cylinder summary information */
/* these fields are cleared at mount time */
char    fs_fmod;      /* super block modified flag */
char    fs_clean;     /* file system is clean flag */
char    fs_ronly;     /* mounted read-only flag */
char    fs_flags;     /* currently unused flag */
char    fs_fsmnt[MAXMNTLEN]; /* name mounted on */

```



## Anhang 5: Dateiformate

```

/* these fields retain the current block allocation info */
long   fs_cgrotor;      /* last cg searched */
struct csum *fs_csp[MAXCSBUFS]; /* list of fs_cs into buffers */
long   fs_cpc;         /* cyl per cycle in postbl */
short  fs_postbl[MAXCPG] [NRPOS]; /* head of blocks */
                                   /* for each rotation */
long   fs_magic;       /* magic number */
u_char fs_rotbl[1];    /* list of blocks for each rotation */
/* actually longer */
};

```

Jedes Plattenlaufwerk enthält eine gewisse Anzahl Dateisysteme. Ein Dateisystem besteht aus einer Anzahl Zylindergruppen. Jede Zylindergruppe besitzt I-Knoten und Daten.

Ein Dateisystem wird von seinem Superblock beschrieben, der wiederum die Zylindergruppen beschreibt. Der Superblock enthält kritische Angaben und wird daher in jeder Zylindergruppe wiederholt, um Datenverluste mit katastrophalen Folgen zu verhindern. Dies geschieht beim Anlegen der Dateisysteme. Die kritischen Daten im Superblock ändern sich nicht, so daß die Kopien nicht weiter angesprochen werden müssen, außer wenn ein Unglück geschieht.

Die in den I-Knoten gespeicherten Adressen können Fragmente von Blöcken adressieren. Dateisystemblöcke mit einer Größe von maximal MAXBSIZE können wahlweise in 2, 4 oder 8 jeweils einzeln adressierbare Teile unterteilt werden; diese Teile können die Größe DEV\_BSIZE oder ein ganzes Vielfaches von DEV\_BSIZE haben.

Große Dateien bestehen ausschließlich aus großen Datenblöcken. Um eine unangemessene Vergeudung von Plattenspeicherplatz zu vermeiden, werden dem letzten Datenblock einer kleinen Datei nur so viele Fragmente eines großen Blocks wie nötig zugeordnet. Das Dateisystemformat unterhält nur einen einzigen Zeiger auf ein solches Fragment, bei dem es sich um einen Teil eines großen unterteilten Blockes handelt. Die Größe eines solchen Fragments ist aufgrund von Informationen im I-Knoten mit Hilfe des Makros blksize(fs, ip, lbn) feststellbar.

Das Dateisystem merkt sich die Verfügbarkeit von Speicherplatz auf Fragmentenebene; um die Verfügbarkeit von Blöcken festzustellen, werden ausgerichtete Fragmente untersucht.

---

## Anhang 5: Dateiformate

---

Der Root-I-Knoten ist die Wurzel (root) des Dateisystems. I-Knoten 0 kann nicht für normale Zwecke verwendet werden, und mit I-Knoten 1 wurden ursprünglich fehlerhafte Blöcke verknüpft; daher ist I-Knoten 2 die Wurzel. (I-Knoten 1 hat zwar heute nicht mehr die ursprüngliche Funktion, aber viele Dump-Bänder gehen noch davon aus, so daß wir davon nicht abweichen können). Dem Verzeichnis lost+found wird der nächste verfügbare I-Knoten zugeordnet, wenn dieses anfangs von mkfs angelegt wird.

Fs minfree gibt den kleinsten noch zulässigen Prozentsatz an Dateisystemblöcken, die frei sein können, an. Wenn die Liste der freien Blöcke weniger als diesen Wert enthält, kann nur der Superuser noch weitere Blöcke zuordnen. Man kann diesen Wert auf 0 setzen, wenn man keine Reserve an freien Blöcken für notwendig hält. Es tritt jedoch ein schwerwiegender Leistungsabfall ein, wenn das Dateisystem zu mehr als 90 % belegt ist; der Standardwert für fs minfree ist daher 10 %.

Erfahrungsgemäß ergibt sich der beste Mittelweg zwischen der Blockfragmentierung und der gesamten Plattenausnutzung bei einer Belastung von 90 % bei einer Fragmentierung von vier. Daher beträgt die standardmäßige Fragmentgröße ein Viertel der Blockgröße.

Für Zylindergruppen gelten folgende Werte und Grenzen: Jeder Zylinder führt Buch über die verfügbaren Blöcke bei unterschiedlichen Drehpositionen, so daß sequentielle Blöcke mit geringstmöglicher Latenzzeit angelegt werden können. NRPOS ist die Anzahl der unterschiedlichen Drehpositionen. Bei NRPOS 8 beträgt die Auflösung der Blockinformationen 2 ms bei einem typischen Plattenlaufwerk mit 3600 Umdrehungen pro Minute.

Fs rotdelay gibt die Mindestdauer in Anzahl Millisekunden bis zur Einleitung eines weiteren Plattentransfers auf demselben Zylinder an. Damit wird der hinsichtlich der Plattenrotation optimale Aufbau der Plattenblöcke in einer Datei festgelegt. Der Standardwert für fs rotdelay ist 2 ms.

Jedes Dateisystem besitzt eine fest zugeordnete Anzahl I-Knoten. Pro NBPI Bytes Plattenspeicherplatz wird ein I-Knoten zugeordnet. Die Zuordnungsstrategie für die I-Knoten ist äußerst konservativ.

## Anhang 5: Dateiformate

MAXIPG begrenzt die Anzahl der I-Knoten pro Zylindergruppe und dient nur dazu, die Struktur möglichst einfach zu gestalten, indem nur ein einziges Element variabler Größe (die „free bit map“) vorgesehen wird.

Anmerkung: MAXIPG muß ein ganzes Vielfaches von INOPB(fs) sein.

MINBSIZE ist die kleinste zulässige Blocklänge. Bei einer MINBSIZE von 4096 Bytes ist es möglich, Dateien von  $2^{32}$  mit nur zwei Indirektionsstufen anzulegen. MINBSIZE muß so groß sein, daß es einen Zylindergruppenblock aufnehmen kann; daher müssen Änderungen bei (struct cg) so klein bleiben, daß die Größe innerhalb von MINBSIZE bleibt. MAXCPG ist nur auf die Dimension eines Array in (struct cg) begrenzt; dieser Wert kann größer gewählt werden, solange die Größe der Struktur innerhalb der von MINBSIZE bleibt. Superblöcke sind nie größer als SBSIZE.

Der Pfadname, auf dem das Dateisystem eingehängt ist, wird in fs fsmnt geführt. MAXMNTLEN definiert den Speicherplatz, der im Superblock für diesen Namen reserviert ist. Der maximale Umfang der zusammenfassenden Informationen pro Dateisystem ist durch MAXCSBUFS definiert. Dieser Wert ist derzeit durch Parameter auf maximal zwei Millionen Zylinder begrenzt.

Pro Zylindergruppe werden die Informationen in Blöcken zusammengefaßt, die aus den Datenblöcken der ersten Zylindergruppe entnommen werden. Diese Blöcke werden von fs\_csaddr (Größe fs\_cssize) zusätzlich zum Superblock eingelesen.

Anmerkung: Sizeof (struct csum) muß eine Potenz von zwei sein, damit das Makro fs\_cs einwandfrei arbeitet.

## Anhang 5: Dateiformate

---

Superblock für ein Dateisystem: MAXBPC begrenzt die Größe der Tabellen für den rotationsbezogenen Aufbau und wird durch den Umstand begrenzt, daß der Superblock die Größe SBSIZE hat. Die Größe dieser Tabellen ist der Blocklänge des Dateisystems umgekehrt proportional. Die Größe der Tabelle nimmt zu, wenn die Sektorlängen keine Potenzen von zwei sind, da dadurch die Anzahl der Zylinder, die vor der Wiederholung des rotationsbezogenen Musters auftreten (`fs_cpc`), steigt. Die Größe der Tabellen für den rotationsbezogenen Aufbau wird abgeleitet von der Anzahl der verbleibenden Bytes in (`struct fs`).

MAXBPG begrenzt die Anzahl der Datenblöcke pro Zylindergruppe und wird von dem Umstand begrenzt, daß Zylindergruppen maximal einen Block groß sein können. Die Größe der Tabelle der freien Blöcke wird von der Größe der Blöcke und der Anzahl der verbleibenden Bytes in der Zylindergruppenstruktur (`struct cg`) abgeleitet.

I-Knoten: Der I-Knoten bildet den Mittelpunkt sämtlicher Dateiaktivitäten im UNIX-Dateisystem. Für jede aktive Datei, jedes aktuelle Verzeichnis, jede eingehängte Datei, Textdatei und die Wurzel wird ein eindeutiger I-Knoten zugeordnet. Ein I-Knoten wird durch Gerät/I-Nummer benannt. Weitere Informationen finden Sie in der include-Datei `<sys/inode.h>`.

## Anhang 5: 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:

- |                      |  |
|----------------------|--|
| <i>label</i>         | 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).  |
| <i>initial-flags</i> | Diese Flags sind die anfänglichen Werte des Systemaufrufs 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 5: Dateiformate

---

- final-flags* 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.
- login-prompt* 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.
- next-label* 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 5: Dateiformate

---

DATEIEN:

/etc/gettydefs

SIEHE AUCH:

Administrator-Kommando: getty

Geräte-datei: termio

Benutzerkommando: login

Systemaufruf: ioctl

## Anhang 5: 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; jeder 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 5: 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 die Leitungen, für die gettys erzeugt werden, 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, wobei die ID immer eindeutig sein muß.

Die meisten Einträge in inittab bewirken, daß ein oder mehrere Prozesse von init gestartet werden. Die wichtigste Ausnahme von dieser Regel ist der Eintrag für einen remote Prozeß. Der Zweck des Eintrags für den remote Prozeß besteht darin, eine Abrechnungs-ID bereitzustellen, die von remote (über einen lokalen Dämon) eingeleiteten Login-Prozessen so benutzt wird, als ob sie von init gestartet worden wären. Die utmp-Abrechnung für remote Login-Prozesse wird effektiv von init in Zusammenarbeit mit dem lokalen Dämonprozeß durchgeführt.

Es folgt das Beispiel eines Eintrags für einen remote Prozeß:

rp0:r:remote:daemon: tty0

Eine typische inittab-Datei enthält einen Eintrag für jedes Pseudo-Terminal, das als Login-Terminal benutzt werden könnte.

---

## Anhang 5: Dateiformate

---

Die Einschränkungen für die möglichen Inhalte der verschiedenen Felder eines Eintrages für einen remote Prozeß werden im folgenden bei der allgemeinen Beschreibung der Felder angegeben.

- 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 `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 markiert ist, wenn ihre Leitung vollständig aus `/etc/inittab` gelöscht ist oder wenn `init` in den `SINGLE USER`-Status übergeht.

**Anhang 5: Dateiformate**

<i>Aktion</i>	Schlüsselwörter in diesem Feld teilen init mit, wie der im Prozeß-Feld angegebene Prozeß zu behandeln ist. Init kennt folgende Aktionen:
remote	Dieser Wert wird für die Abrechnung beim remote Login verwendet. Diese ID kann als Login-Eintrag von Dämonen für eine remote Anmeldung verwendet werden. Wird dieser Eintrag mit dem Wert off markiert, so ist dieser Eintrag deaktiviert.
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.
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.

© Weitergabe sowie Vervielfältigung dieses Urtextes, Vorwertung und Mitteilung ihres Inhalts nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zu Schadenersatz. Alle Rechte für den Fall der Patenterteilung oder Gebrauchsmusterteilung vorbehalten.

**A5**

---

## Anhang 5: Dateiformate

---

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 Systemaufruf 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.
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.

## Anhang 5: Dateiformate

**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 auch `s` angegeben werden. Dies bedeutet, 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.

Bei Einträgen für remote Prozesse muß dieses Feld mit dem Schlüsselwort `daemon` beginnen. Der zweite Teil dieses Feldes ist die letzte Komponente des vollständigen Pfadnamens des remote Terminals. Der Eintrag für `/dev/tty5` hätte z. B. folgende Form:

`rp5:r:remote:daemon tty5`

## Anhang 5: Dateiformate

---

### HINWEISE:

Das ID-Feld der Einträge für remote Prozesse hat keine Besonderheiten. Es muß wie jede andere ID eindeutig sein.

Wird das Aktions-Feld für einen remote Prozeß von remote in off abgeändert, so wird das Pseudo-Terminal als Login-Terminal deaktiviert. Um Änderungen in Einträgen für remote Prozesse vorzunehmen, muß ebenso wie bei Änderungen in anderen Einträgen „telinit q“ oder „kill -HUP 1“ angegeben werden.

### DATEIEN:

/etc/inittab

### SIEHE AUCH:

Administrator-Kommandos: getty, init  
Benutzerkommandos: sh, who  
Systemaufrufe: exec, open, signal

## Anhang 5: 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 in <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:

Dateiformat: fs  
Systemaufruf: stat

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

## Anhang 5: 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 5: Dateiformate

---

### mtab – Tabelle der eingehängten Dateisysteme

#### SYNTAX:

```
#include <fstab.h>
#include <mtab.h>
```

#### BESCHREIBUNG:

Mtab liegt im Verzeichnis /etc und enthält eine Tabelle der vom Kommando mount eingehängten Geräte. Mit umount werden die Einträge gelöscht.

Die Tabelle besteht aus einer Reihe von mtab-Strukturen, wie sie in <mtab.h> definiert sind. Jeder Eintrag enthält den durch Leerzeichen aufgefüllten Pfadnamen wo die Gerätedatei eingehängt ist, den durch Leerzeichen aufgefüllten Namen der Gerätedatei und eins der in <fstab.h> definierten Typfelder. Von der Gerätedatei sind alle zugehörigen Verzeichnisse entfernt, d. h. alles bis zum letzten '/' ist weggefallen. Das Typfeld gibt den Zugriffsschutz der Dateisysteme an.

#### DATEIEN:

/etc/mtab

#### SIEHE AUCH:

Administrator-Kommando: mount

---

## Anhang 5: 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 5: 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 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 5: 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 22
# Information über eingehende Post
MAIL=/usr/mail/myname
# 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 5: Dateiformate

### security – Login-Sicherheitsmaßnahmen

#### BESCHREIBUNG:

Das Programm /bin/login stellt in beiden Welten mehrere Sicherheitsmaßnahmen zur Verfügung:

- Paßwortänderung in gewissen Zeitabständen (nur System V).
- 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 /usradm/badlog aufgezeichnet.
THREE_TRYS	Die Anzahl der nicht erfolgreichen Anmeldungen wird auf drei begrenzt.
DIALPASS	Für Wählverbindungen wird ein zusätzliches Paßwort verlangt.

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

## Anhang 5: Dateiformate

---

Das Paßwort für Wählverbindungen wird verschlüsselt und in der Datei `/etc/d_passwd` abgelegt. Ist diese Datei nicht vorhanden, findet keine Überprüfung des Paßworts für Wählverbindungen statt.

DATEIEN:

`/etc/security`

SIEHE AUCH:

Benutzerkommando: `login`

## Anhang 5: 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 dasselbe Terminal werden durch mehrfache Verweise (Links) auf dieselbe 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 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 5: 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  $256 \cdot \text{second} + \text{first}$ .) 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 derselben 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  $\sim X$  oder  $\backslash c$  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 5: 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 setufterm andere Funktionsmöglichkeiten erwartet als in der Datei effektiv vorhanden sind. Dies kann der Fall sein, wenn die Datenbasis seit der Rekompilierung von setufterm 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 setufterm 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, cudl=^J, ind=^J, bel=^G, am, cubl=^H,
  ed=^, el=^^, clear=^L, cup=^T^p1^c^p2^c,
  cols#80, lines#24, cuf1=^X, cuul=^Z, home=^_),

000 032 001      \0 025 \0 \b \0 212 \0  " \0  m  i  c  r
020  o  t  e  r  m  i  a  c  t  4  |  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
540 377 377 377 377 377 377 007 \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
```

Die kompilierten Einträge dürfen insgesamt nicht länger als 4096 Bytes sein. Das Namensfeld kann maximal 128 Bytes lang sein.

**A5**

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

## Anhang 5: Dateiformate

---

DATEIEN:

/usr/lib/terminfo/\*\* kompilierte Datenbasis der Terminalfunktionen

SIEHE AUCH:

Dateiformat: terminfo

## Anhang 5: Dateiformate

### terminfo – Datenbasis der Terminalfunktionen

#### BESCHREIBUNG:

Terminfo ist eine Datenbasis, in der Terminals beschrieben sind. Diese Daten werden beispielsweise von vi und curses benutzt werden. 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. dap4x. 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 5: 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 terminfo-Ebene) die Funktionsmöglichkeit anspricht. Der capname (Name für die Funktionsmöglichkeit) ist die im Text der Datenbasis verwendete Kurzbezeichnung. Dieser Name wird von demjenigen verwendet, der die Datenbasis aktualisiert. Der i.code ist der aus zwei Buchstaben bestehende 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 (#<sub>i</sub>).
- (\*) bedeutet, daß ein Auffüllen entsprechend der Anzahl der betroffenen Zeilen erfolgen kann.
- (#<sub>i</sub>) gibt den i-ten Parameter an.

## Anhang 5: Dateiformate

Variable Booleans	Cap- Name	I. Code	Beschreibung
auto_left_margin,	bw	bw	cub1 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 5: Dateiformate

cursor_down,	cu d1	do	Down one line
cursor_home,	home	ho	Home cursor (if no cup)
cursor_invisible,	civis	vi	Make cursor invisible
cursor_left,	cub1	le	Move cursor left one space
cursor_mem_address,	mrcup	CM	Memory relative cursor addressing
cursor_normal,	cnorm	ve	Make cursor appear normal (undo vs/vi)
cursor_right,	cuf1	nd	Non-destructive space (cursor right)
cursor_to_ll,	ll	ll	Last line, first column (if no cup)
cursor_up,	cuu1	up	Upline (cursor up)
cursor_visible,	cvvis	vs	Make cursor very visible
delete_character,	dch1	dc	Delete character (P*)
delete_line,	dl1	dl	Delete line (P*)
dis_status_line,	dsl	ds	Disable status line
down_half_line,	hd	hd	Half-line down (forward 1/2 linefeed)
enter_alt_charset_mode,	smacs	as	Start alternate character set (P)
enter_blink_mode,	blink	mb	Turn on blinking
enter_bold_mode,	bold	md	Turn on bold (extra bright) mode
enter_ca_mode,	smcup	ti	String to begin programs that use cup
enter_delete_mode,	smdc	dm	Delete mode (enter)
enter_dim_mode,	dim	mh	Turn on half-bright mode
enter_insert_mode,	smir	im	Insert mode (enter);
enter_protected_mode,	prot	mp	Turn on protected mode
enter_reverse_mode,	rev	mr	Turn on reverse video mode
enter_secure_mode,	invis	mk	Turn on blank mode (chars invisible)
enter_standout_mode,	smso	so	Begin stand out mode
enter_underline_mode,	smul	us	Start underscore mode
erase_chars,	ech	ec	Erase #1 characters (PG)
exit_alt_charset_mode,	rmacs	ae	End alternate character set (P)
exit_attribute_mode,	sgr0	me	Turn off all attributes
exit_ca_mode,	rncup	te	String to end programs that use cup
exit_delete_mode,	rmdc	ed	End delete mode
exit_insert_mode,	rmir	ei	End insert mode
exit_standout_mode,	rmso	se	End stand out mode
exit_underline_mode,	rmul	ue	End underscore mode
flash_screen,	flash	vb	Visible bell (may not move cursor)
form_feed,	ff	ff	Hardcopy terminal page eject (P*)
from_status_line,	fsl	fs	Return from status line
init_1string,	is1	i1	Terminal initialization string
init_2string,	is2	i2	Terminal initialization string
init_3string,	is3	i3	Terminal initialization string
init_file,	if	if	Name of file containing is
insert_character,	ich1	ic	Insert character (P)
insert_line,	il1	al	Add new blank line (P*)
insert_padding,	ip	ip	Insert pad after character inserted (p*)
key_backspace,	kbs	kb	Sent by backspace key
key_catab,	ktbc	ka	Sent by clear-all-tabs key
key_clear,	kclr	kC	Sent by clear screen or erase key

## Anhang 5: Dateiformate

key_ctab,	kctab	kt	Sent by clear-tab key
key_dc,	kdch1	kD	Sent by delete character key
key_dl,	kdll	kL	Sent by delete line key
key_down,	kcud1	kd	Sent by terminal down arrow key
key_eic,	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,	kil1	kA	Sent by insert line
key_left,	kcub1	kl	Sent by terminal left arrow key
key_ll,	kll	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,	kcuu1	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,	rmm	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 5: 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
ptr_off,	mc4	pf	Turn off the printer
ptr_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,	ipro	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
ptr_non,	mc5p	pO	Turn on the printer for #1 bytes



## Anhang 5: 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$!$' '%&#&#p2$' '%&#&c,
  cuu1=~\E; , cvvis=~\EW, db, dch1=~\E^A$<16*>, dim=~\EE, dl1=~\E^B$<3*>,
  ed=~\E^C$<16*>, el=~\E^U$<16>, eo, flash=~\Ek$<20>\EK, ht=~\t$<8>,
  ili=~\E^R$<3*>, in, ind=~^J, .ind=~^J$<9>, ip=$<16*>,
  is2=~\EU\Ef\E7\E5\E8\E1\ENH\EK\E\200\Eo6\200\Eo\47\E,
  kbs=~^h, kcub1=~\E, kcu1=~\E<, kcu1=~\E=, kcuu1=~\E; ,
  kf1=~\E5, kf2=~\E6, kf3=~\E7, khome=~\E7,
  lines#24, mir, pb#9600, prot=~\EI, rep=~\Er$!$&#p2$' '%&#&c< 2*>,
  rev=~\ED, rmcup=~\Ev $<6>\Ep\r\n, rmir=~\E\200, rmkx=~\Ex,
  rmso=~\Ed\Ee, rmul=~\Eg, rmul=~\Eg, sgr0=~\E\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 Funktionsmöglichkeit am angegeben. Bei numerischen Funktionsmöglichkeiten wird hinter dem Namen das Zeichen # und dann der Wert angegeben.

---

## Anhang 5: Dateiformate

---

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 zehnte! 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.

## Anhang 5: 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 Umgebungsvariable 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 5: 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. `cud1` 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 dieselbe 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 5: 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 mrcup angegeben werden.

Der Parametermechanismus erzeugt 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:

%%	gibt % aus
%d	pop() ausgeben wie bei printf
%2d	pop() ausgeben wie bei %2d
%3d	pop() ausgeben wie bei %3d
%02d	
%03d	wie bei printf
%c	pop() ausgeben ergibt %c
%s	pop() ausgeben ergibt %s

---

## Anhang 5: Dateiformate

---

%p[1-9]	i-ten Parameter im Speicher ablegen
%P[a-z]	Variable [a-z] auf pop() setzen
%g[a-z]	Variable [a-z] holen und im Speicher ablegen
%'c'	Zeichenkonstante c
%{nn}	Ganzzahl-Konstante nn
%	%- %* %/ %m
%& %  %~	Bit-Operationen: push(pop() op pop())
%= %> %<	logische Operationen: push(pop() op pop())
%ö %~	unäre Operationen push(pop() op pop())
%i	1 zu den ersten beiden Parametern addieren (bei ANSI-Terminals)
%? expr %t thenpart %e elsepart%;	if-then-else, %e elsepart ist wahlfrei. else-if's sind möglich wie bei Algol 68: %? c <sub>1</sub> %t b <sub>1</sub> %e c <sub>2</sub> %t b <sub>2</sub> %e c <sub>3</sub> %t b <sub>3</sub> %e c <sub>4</sub> %t b <sub>4</sub> %e %; c <sub>i</sub> sind Bedingungen, b <sub>i</sub> 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 5: 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 vt100), kann das Kommando zum Einstellen dieses Bereiches mit der

---

## Anhang 5: Dateiformate

---

Funktionsmöglichkeit `csr` beschrieben werden; `csr` besitzt zwei Parameter: die oberste und die unterste Zeile des Rollbereiches. Die Cursorposition ist nach diesem Kommando leider nicht definiert. Man kann mit diesem Kommando die Wirkung des Einfügens bzw. Löschens 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/Löschens von Zeichen zwei Grundtypen zu unterscheiden, die in `terminfo` beschrieben werden können. Bei der häufigsten Form des Einfügens/Löschens 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 Bildschirm löschen und dann durch Cursorbewegungen getrennte Textteile schreiben. Schreiben Sie `abc def`, wobei zwischen dem `abc`



## Anhang 5: Dateiformate

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 hinübergeschoben 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 angegeben. Die Sequenz, mit der der Einfügemodus verlassen wird, wird mit rmir angegeben. 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.

© „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 Gebrauchsmusterantragung vorbehalten.“

---

## Anhang 5: Dateiformate

---

Wenn man im Einfügemodus arbeitet, muß man hin und wieder Zeichen in derselben 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` weggelassen, 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 verfügt, mit dem das aktuelle Zeichen unterstrichen und der Cursor um

## Anhang 5: Dateiformate

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 Modes 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 Modes 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 Modes 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 `rmcup` angegeben werden. Dies trifft beispielsweise bei Terminals wie dem Concept zu, die mehr als eine Seite Bildschirmspeicher

---

## Anhang 5: Dateiformate

---

besitzen. Wenn das Terminal nur speicherbezogene Cursoradressierung und keine bildschirmbezogene Cursoradressierung besitzt, muß ein ein Bildschirm großes Fenster 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 löscher 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 kcub1, kcuf1, kcuu1, kcucl1 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), ktabc (alle Tabs löschen), kctab (Tabstop in dieser Spalte löschen), kclr (Bildschirm löschen oder Erase-Taste), kdch1 (Zeichen löschen), kdll1 (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 (vorangehende 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 Tasten mit ka1, ka3, kb2, kc1 und kc3 angegeben werden. Diese Tasten sind nützlich, wenn die Funktionen eines richtungsorientierten

---

## Anhang 5: Dateiformate

---

Cursorsteuertastenfildes 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 Verdopplung der Strings eingerichtet werden, wenn man die in jedem Fall geltenden Sequenzen in is2 und die Sonderfälle in is1 und is3 codiert. 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 wer-

## Anhang 5: Dateiformate

---

den 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 5: 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 derselben 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 `wsl` 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.

A5

© „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 Gebrauchsmustereintragung vorbehalten.“

---

## Anhang 5: 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



## Anhang 5: Dateiformate

10) und die Zeichenkette, mit der diese Taste zu programmieren ist. Mit Funktionstastennummern außerhalb dieses Bereiches können unter Umständen nicht definierte Tasten terminalabhängig programmiert werden. Der Unterschied zwischen diesen Funktionsmöglichkeiten ist folgender: pfkey bewirkt, daß das Betätigen der betreffenden Taste dieselbe Wirkung hat wie wenn der Benutzer die entsprechende Zeichenfolge eintastet; pfloc bewirkt, daß die Zeichenfolge vom Terminal im Lokalmodus ausgeführt wird; pfx 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.

### 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 angezogenen Terminaltyp angegebenen Funktionsmöglichkeiten außer

## Anhang 5: Dateiformate

---

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 5: Dateiformate

### u\_ttytype – Terminal-Einstellungen

SYNTAX:

*Symbol ttyxx*

BESCHREIBUNG:

Die Datei u\_ttytype wird vom Programm /bin/login von 4.2 BSD benutzt. Jeder Eintrag in der Datei dient dazu, eine Terminal-Einstellung in der Datei /etc/gettydefs von System V nachzuschlagen. Ein Eintrag in der Datei u\_ttytype sieht wie folgt aus:

*Symbol ttyxx*

*Symbol* ist eine symbolische Beschreibung der Terminal-Modes. Die Symbole in der Datei /etc/gettydefs kommen aus der Gerätedatei termio in den Terminal-Schnittstellen von System V.

HINWEIS:

Die Datei u\_ttytype darf keine Leerzeilen und keine Kommentarzeilen enthalten.

SIEHE AUCH:

Administrator-Kommando: getty

Dateiformat: gettydefs

Gerätedatei: termio

## Anhang 5: Dateiformate

---

### **u\_universe – Datei zur Einstellung des Standarduniversums**

#### SYNTAX:

*Login-Name:Universum*

#### BESCHREIBUNG:

Die Datei u\_universe enthält für jeden Benutzer den Login-Namen und das Universum, in dem er sich nach der Anmeldung befindet.

U\_universe ist eine ASCII-Datei. Jeder Benutzereintrag in der Datei enthält zwei Felder, die durch einen Doppelpunkt getrennt sind. Jeder Benutzereintrag ist vom nächsten durch ein Zeilenende-Zeichen getrennt. Die Datei liegt im Verzeichnis /etc/u\_universe und ist mit allgemeiner Leseerlaubnis ausgestattet.

#### DATEIEN:

/etc/u\_universe

#### SIEHE AUCH:

Dateiformat: passwd

Benutzerkommandos: login, universe

## Anhang 5: 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, haben die folgende von <utmp.h> definierte Struktur:

```
#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 */
};

/* 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
```

---

## Anhang 5: Dateiformate

---

```
#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 6: Stichwortverzeichnis

### A6 Stichwortverzeichnis

.profile 11.7

/dev/init 10.1

/etc/bcheckrc 10.2.1

/etc/brc 10.2.1

/etc/cshinit 10.1

/etc/cshinit.att 10.5

/etc/cshinit.ucb 10.5

/etc/d\_passwd 10.4

/etc/fstab A3

/etc/gettydefs 10.1

/etc/gettytab 10.1

/etc/group 11.8

/etc/halt 8.4

/etc/inittab 10.1

/etc/mnttab A3

/etc/motd 11.5

/etc/passwd 11.8

/etc/profile 11.7

/etc/rc 10.2.1

/etc/reboot 8.4

/etc/security 10.1, 10.4

/etc/shutdown 8.4

/etc/ttys 10.1

/etc/ttytype 10.1

/etc/utmp 10.1

/etc/u\_ttytype 10.1

/etc/u\_universe 9, 10.1

/etc/wtmp 10.1

/usr-Dateisystem 7.1

/usr/adm/badlog 10.4

/usr/adm/cronlog 11.4

/usr/adm/errfile 11.4

/usr/adm/pacct 11.4

/usr/adm/wtmp 10.1, 11.4

---

## Anhang 6: Stichwortverzeichnis

---

/usr/lib/crontab 11.9  
/usr/news 11.5  
/usr/spool 11.4

4.2 BSD-Kernel 9

Abbrechen, aktive Prozesse (killall) A3  
Abrechnungen, täglich (runacct) A3  
Abrechnungsdateien addieren (acctmerge) A3  
Abrechnungsdateien zusammenfassen (acctmerge) A3  
Abrechnungssätze bearbeiten (fwtmp) A3  
accept 12.2.2, A3  
Accounting-Kommandos (acct) A3  
Accounting-Prozeduren (acctsh) A3  
Accounting-Überblick (acct) A3  
acct A3, A5  
acctcms A3  
acctcon 11.4, A3  
acctcon1 A3  
acctcon2 A3  
acctdisk A3  
acctdusg A3  
acctmerge A3  
accton A3  
acctprc A3  
acctprc1 A3  
acctprc2 A3  
acctsh A3  
acctwtmp A3  
Addieren, Abrechnungsdateien (acctmerge) A3  
AGEING 10.4  
aktive Prozesse abbrechen (killall) A3  
Anlegen, FIFO-Datei (mknod) A3  
Anlegen, Gerätedatei (mknod) A3  
Anlegen, Named-Pipe (mknod) A3  
att 9.1  
Aufbauen, Dateisystem (mkfs) A3  
Auflisten, Statistik Dateisystem (ff) A3  
Auftrags-ID 12.2.1



---

**Anhang 6: Stichwortverzeichnis**

---

Ausführen, tägliche Abrechnungen (runacct) A3  
 Ausführungseinheit 2.2.1  
 Ausgabeanforderung 12.1  
 Aushängen, Dateisystem (umount) A3  
 Auswerten, protokollierter Fehler (errpt) A3  
 Auswerten, Systemaktivitäten (sar) A3  
 Autoboot-Status 4.2.2

Bandeinfädeler, automatische A2  
 Baud-Rate 12.6.2  
 bcheckrc A3  
 Bearbeiten, Abrechnungssätze (fwtmp) A3  
 Beenden, Fehlerprotokollierungs-Dämon (errstop) A3  
 Begin of Tape A2  
 Benutzerkommandos 12.2.1  
 Benutzerstatistik 11.9  
 Benutzerumgebung 11.7  
 Bereinigen, uuop-Spool-Verzeichnis (uuclean) A3  
 Betriebssysteminstallation 7  
 Board 7.2.2  
 Boot-Gerät 4.2.2  
 Boot-Optionen 4.2.2  
 Bootblock 5.7  
 BOT-Marke A2  
 brc A3

C.E.ON 5.1  
 Cache-Streamer-Tape 2.5  
 cancel 12.2.1  
 CE-Betriebsart 3.2  
 CE-Schlüssel 7.2  
 chargefee A3  
 chroot 11.1, A3  
 ckpacct A3  
 clri A3  
 Console Operating System 3  
 Control Store 4.2.5  
 Controllernummer 7.2.2  
 COS 3

---

## Anhang 6: Stichwortverzeichnis

---

COS-Release 4.2.2  
COS-Schlüssel, aktueller 5.6  
cpio A5  
cpio-Format 7.1  
cron 11.4, 11.9, A3  
crontab 11.4  
csh 10.5

Dämon-Prozesse 10.1  
Dateien, systemspezifische 7.2.4  
Dateistruktur 9  
Dateisystem aufbauen (mkfs) A3  
Dateisystem aushängen (unmount) A3  
Dateisystem einhängen (mount) A3  
Dateisystem-Statistik auflisten (ff) A3  
Dateisysteme kopieren (volcopy) A3  
Dateisystemtabelle A3  
Datenschutzmechanismen 11.1  
Datensicherung, inkrementelle (finc) A3  
Datenstrukturen, interne 8.5  
dd-Format 7.1  
devnm A3  
df 11.6, A3  
Diagnoseroutinen 3.1  
Diagnostetests 6.2  
DIALPASS 10.4  
disable 12.2.1  
Diskettenlaufwerkstatus 5.4  
dk 7.2.2  
dodisk A3  
Druckaufträge freigeben (accept) A3  
Druckaufträge sperren (reject) A3  
Drucker 12.1  
Drucker löschen 12.4.5  
Druckerkonfiguration 12.4.1  
Druckername 12.4.1.1  
Druckerspezifizierung 12.4.2  
du 11.6  
Duplex-Betrieb 4.3.1

## Anhang 6: Stichwortverzeichnis

Eingabepuffer 3.2.5  
 Einhängen, Dateisystem (mount) A3  
 Einstellen, Geschwindigkeit (getty) A3  
 Einstellen, Leitungsprozedur (getty) A3  
 Einstellen, Terminalmodi (getty) A3  
 Einstellen, Terminaltyp (getty) A3  
 Einzelschrittverarbeitung 5.8  
 enable 12.2.1  
 Ermitteln, Geräte name (devnm) A3  
 err A4  
 errdead A3  
 errdemon A3  
 errdemon-Mechanismus A1  
 errfile A5  
 errpt A1, A3  
 errstop A3  
 Execution-Unit 2.2.1  
 Extrahieren, Fehlersätze (errdead) A3  
  
 Fehler-Code A1  
 Fehlermeldungen 12.5  
 Fehlerprotokollierungs-Dämon A3  
 Fehlerprotokollierungs-Dämon beenden (errstop) A3  
 Fehlersätze extrahieren (errdead) A3  
 Fehlersatz A3  
 ff A3  
 FIFO-Datei anlegen (mknod) A3  
 filesave A3  
 find A3  
 find 11.6  
 find -cpio 11.2.2  
 format A3  
 Frame-Status 3.2.1  
 freq A3  
 freie Plattenblöcke melden (df) A3  
 fs A5  
 fsck 11.2.1, 8.4, A3  
 Füllgrad 11.6  
 Funktionsauswahl 3.2.3

© „Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und  
 Mitteilung ihres Inhalts in jeglicher Weise ist ohne schriftliche Genehmigung  
 der Nixdorf Computer AG. Alle Rechte für den Fall  
 der Patenterteilung oder Gebrauchsmustereintragung vorbehalten.“

---

## Anhang 6: Stichwortverzeichnis

---

fuser A3  
fwtmp A3

Generieren, UNIX-Kern (makesys) A3  
Geräte-datei anlegen (mknod) A3  
Geräte-dateien 12.1, 12.4.1.1  
Geräte-name ermitteln (devnm) A3  
Geräte-zuordnung 12.4.2  
Geschwindigkeit einstellen (getty) A3  
getty 10, A3  
getty-Aufruf 12.7.1  
gettydefs A5  
group A5  
grpck A3  
Gruppen-datei prüfen (grpck) A3  
Gruppen-zugehörigkeit 11.8

Halb-Duplex-Betrieb 4.3.1  
halt A3  
Hardware-Fehlerprotokolle 11.4  
Haupt-nutzungszeit (acctcon) A3  
Hog-Faktor (acctcms) A3

I-Knoten löschen (clri) A3  
I/O-Fehler A1  
Identifizieren, Prozesse (fuser) A3  
IMPL 6.2  
init 10, A3, A4  
init-Prozeß 10.1  
Initialisieren, Prozeßsteuerung (init) A3  
Initialisierungs-Mikroprogramm 6.2  
Initialisierungsprogramme 6.2  
inittab A5  
Inkonsistenzen 8.5  
inkrementelle Datensicherung (finc) A3  
inode A5  
inst.att 10.6  
inst.ucb 10.6  
install A3

---

**Anhang 6: Stichwortverzeichnis**

---

Installation 12.3  
 Installationsbänder 7  
 Installationskommandos (install) A3  
 Instruction-Unit 2.2.1  
 Instruktionseinheit 2.2.1  
 Interrupt-Status A1  
 IOC-Kanal 7.2.2  
 IOCDISK-Fehler A1  
 issue A5  
 itp A4  
 ITP laden (nlditp) A3  
  
 Kanalnummer 4.2.4, 7.2.2  
 kcore-Minuten (acctcms) A3  
 Kernel-Debugger 4.2.2  
 kill 12.7.1  
 killall A3  
 Klasse 12.1  
 Klassennamen 12.4.1.2  
 kmem A4  
 Kommandozeilen generieren 12.6.1  
 Konfiguration 12.1  
 Konfigurationsdaten 7.2.2  
 Konfigurationskomponente 7.2.2  
 Konfigurationsparameter 7.2.2  
 Konfigurieren, Spool-System (lpadmin) A3  
 Konfigurierung 7.2.2  
 Kopieren, Dateisysteme (volcopy) A3  
  
 labelit A3  
 Laden, ITP (nlditp) A3  
 Laden, Programm (reboot) A3  
 Laden, UNIX-Kern (unixboot) A3  
 lastlogin A3  
 LEDs 5.9  
 Leitungsprotokoll A3  
 Leitungsprozedur einstellen (getty) A3  
 Leuchtanzeige 5.9  
 Leuchtdioden A2

---

**Anhang 6: Stichwortverzeichnis**

---

link A3  
LOAD A2  
Löschen I-Knoten (clri) A3  
Log-Dateien 11.4, 12.5  
login 10  
Login-Accounting 10.1  
LOG\_TO\_FILE 10.4  
lp 12.2.1  
lpadmin 12.2.2, A3  
lpmove 12.2.2, A3  
lpsched 12.2.2, 12.5.1, A3  
lpshut 12.2.2, 12.5.2, A3

Magnetbandgeräte 2.5  
Magnetplattenlaufwerke 2.5  
makesys A3  
Melden, freie Plattenblöcke (df) A3  
mem A4  
Microcode-Sequencing-Unit 2.2.1  
Mikrocode-Release 4.2.2  
Mikrocode-Versionsnummer 7.2.2  
Mikrocode-Verwalter 2.2.1  
mkfs A3  
mknod A3  
Modellschnittstellen 12.4.1.1, 12.6  
monacct A3  
mount A3  
Mount-Tabelle A3  
mtab A5  
Multi-Prozessorumgebung 4.2.2  
mvdire A3

Nachrichtenspeicher 3.3  
Named-Pipe anlegen (mknod) A3  
ncheck A3  
Nebennutzungszeit (acctcon) A3  
newgrp 11.8  
news 11.5  
nlditp A3

## Anhang 6: Stichwortverzeichnis

Non-Prime (acctcon) A3  
 null A4  
 nulladm A3

ON-LINE A2  
 Online-Plattenformatierer (format) A3  
 OPERATOR-Betriebsart 3.2

Parameter, nicht-optionale 12.4.1.1  
 Parameter, optionale 12.4.1.2  
 PASSREQ 10.4  
 passwd A5  
 Paßwortänderung 10.4  
 Paßwortdatei prüfen (pwck) A3  
 Paßwortpflicht 10.4  
 Peripheriegerät, physikalisches 12.1  
 Pico Store 4.2.5  
 Plattenbelegung 11.6  
 Plattenblöcke, freie (df) A3  
 Plattennummer 7.2.2  
 Plattenschalter 7.2  
 Plattenverwendung 11.6  
 Power-Schlüssel 7.2  
 powerfail A3  
 prctmp A3  
 prdaily A3  
 Prime (acctcon) A3  
 profile A5  
 Programm laden (reboot) A3  
 PROTECT 7.2  
 Prozeß-Abrechnungssätze zusammenfassen (acctcms) A3  
 Prozeß-Accounting 11.4  
 Prozeßabrechnung (acctprc) A3  
 Prozesse abrechnen (killall) A3  
 Prozesse identifizieren (fuser) A3  
 Prozeßsteuerung initialisieren (int) A3  
 prtacct A3  
 Prüfen, Dateisysteme (fsck) A3  
 Prüfen, Gruppendatei (grpck) A3

---

## Anhang 6: Stichwortverzeichnis

---

Prüfen, Paßwortdatei (pwck) A3  
Prüfprogramm 8.4  
Pseudo-Tty 10.1  
pwck A3

Raw-Mode A3  
rc A3  
reboot A3  
Rechnerstop (halt) A3  
reject 12.2.2, A3  
Rekonstruieren, Sicherungsband (frec) A3  
Remote-Konsole 4.1  
Remote-Prozesse 10.1  
Reparieren, Dateisysteme (fsck) A3  
RESET-Schalter 6  
Root-Dateisystem 7.1  
Root-Partition (format) A3  
Rückgabecode 12.6.3  
Rückspulen, automatisches A2  
Rückspulen, manuelles A2  
runacct A3  
Rundschreiben senden (wall) A3

sa1 A3  
sa2 A3  
sadc A3  
sar A3  
SBI 2.2.1  
SCHEDLOCK 12.5.1  
Scheduler 12.1, 12.5  
Scheduler starten 12.5.1  
Scheduler stoppen 12.5.2  
Schnittstellenprogramm 12.1, 12.4.1.1, 12.5, 12.6  
Schreibring A2  
Schutzmechanismen 11.2.1  
security A5  
Senden, Rundschreiben (wall) A3  
serielle Drucker 2.5  
Set-Group-ID-Bit 11.1



## Anhang 6: Stichwortverzeichnis

Set-User-ID-Bit 11.1  
 Set-User-ID-Erlaubnis 12.3.1  
 Set-User-ID-Programme 11.1  
 Shell-Prozeduren, Accounting (acctsh) A3  
 Shell-Prozeduren, Systeminitialisierung (brc) A3  
 shutacct A3  
 shutdown 10, A3  
 Sicherung, 8-wöchige 11.2.1  
 Sicherung, physikalische 11.2.2  
 Sicherung, tägliche 11.2.1  
 Sicherung, wöchentliche 11.2.1  
 Sicherungsband rekonstruieren (frec) A3  
 Sicherungskopien 7.2.5, 11.2  
 Sicherungsmaßnahmen 10.1  
 Sicherungsmedium 11.2.1  
 Single-Prozessorumgebung 4.2.2  
 Speicher-Dump A3  
 Speicherplatz 11.6  
 Spool-System 12  
 Spool-System konfigurieren (lpadmin) A3  
 Spooler-Dateien 12.2.2  
 Spooler-Kommandos 12.2.2  
 Spooler-Konfiguration 12.4  
 Spooler-Verzeichnis 12.3  
 SSP-Fehlerarten 5.2  
 Standard-Ausgabegerät 12.1  
 Standard-Systemdrucker 12.4.4  
 Standardschlüssel 5.6  
 Standardumgebung 10.1, 11.7  
 Start-Scripts 10  
 startup A3  
 Status, rekursiver 5.8  
 Status-Byte A1  
 Statuslogbuch 11.4  
 Statusmeldungen 3.2.1  
 Stoppen, System (shutdown) A3  
 su 11.1, 11.9  
 Subkanalnummer 4.2.4  
 Superuser 12.2.2

---

## Anhang 6: Stichwortverzeichnis

---

Superuser-Paßwort 11.1  
System stoppen (shutdown) A3  
Systemaktivitäten auswerten (sar) A3  
Systembus-Interface 2.2.1  
Systeminitialisierung (brc) A3  
Systemkonsole 3.3  
Systemnachrichtenpuffer 5.2  
Systemstatus-Zeile 3.2.1

Tagesnachrichten 11.5  
tapesave A3  
tar 11.2.2  
tcsh 10.5  
telinit 10, A3  
TERM 10.1  
term A5  
Terminal-Modi 12.6.2  
Terminaleinstellungen 10.1  
Terminalleitung 10.1  
Terminalmodi einstellen (getty) A3  
Terminaltyp einstellen (getty) A3  
Terminalzeiten abrechnen (acctcon) A3  
terminfo A5  
termio A4  
THREE\_TRYIS 10.4  
tty A4  
turnacct A3  
TZ 10.5

ucb 9.1  
Überwachen, uucp-Netz (uusub) A3  
UIB 5.7  
Umbenennen, Verzeichnis (mvdir) A3  
Umgebungsvariablen 10.5  
umount A3  
Unit Initialization Blocks 5.7  
UNIX-Kern generieren (makesys) A3  
UNIX-Kern laden (unixboot) A3  
unixboot A3

---

**Anhang 6: Stichwortverzeichnis**

---

unlink A3  
UNLOAD A2  
Unterkanal 7.2.2  
Ursache-Code A1  
utmp A5  
uuclean A3  
uucp-Netz überwachen (uusub) A3  
uucp-Spool-Verzeichnis bereinigen (uuclean) A3  
uusub A3  
u\_ttytype A5  
u\_universe A5

Verknüpfung, bedingte symbolische 9.1  
Verschieben, Verzeichnis (mmdir) A3  
Verwaltungsdateien 11.4  
Verzeichnis umbenennen (mmdir) A3  
Verzeichnis verschieben (mmdir) A3  
Verzeichnisse verkleinern 11.3  
volcopy 11.2.2, A3

Wählanschluß 11.1  
wall 11.5, A3  
Wechseln, Wurzelverzeichnis (chroot) A3  
Welt, aktuelle 9  
whodo A3  
Wiederholungszähler 5.2  
Winchester Festplatten 2.5  
wtmp A5  
wtmpfix A3  
Wurzelverzeichnis wechseln (chroot) A3

Zeilendrucker 2.5  
Zeit-Daemon (cron) A3  
Zeitzone-Parameter 10.5  
Zusammenfassen, Abrechnungsdateien (acctmrg) A3  
Zwei-Welten-Konzept 9

---

