

SSBMX300I

Kursordner

1 : Kursorganisation

2 : Vorstellung

3 : Installation

4 : SINIX 5.4

5 : Backup

6 : Sytemparameter

7 : Freigabemitteilung

8 : TK -Informationen

9 : Praxis

10 : Anhang

2

Kapitel 2

Kapitel 2 : Vorstellung

MX300-Produktblatt	II-1
SINIX V5.4-Produktvorstellung ..	II-2
UNIX System V.4 : Einführung..	II-3

SIEMENS



MX 300

SINIX Mehrplatzsysteme

Allgemeines

Mit dem SINIX-Mehrplatzsystem MX300 bietet Siemens eine Systemfamilie in unterschiedlichen Leistungs- und Ausbauprodukten an. Mit insgesamt fünf Modellvarianten stellt der MX300 eine ideale Leistungsbandbreite im Bereich der kleinen und mittleren SINIX-Mehrplatzsysteme dar.

Aufgrund seines modularen Aufbaus, seiner konstruktiven Gestaltung als Unterschrankmodell, der Ausbaubarkeit und der starken Systemleistung läßt sich der MX300 gut in jede Bürolandschaft integrieren und den jeweiligen Aufgaben des Anwenders optimal anpassen.

Besondere Merkmale

- Einbettung in die TRANSDATA- und SNA-Netzarchitektur
- Anschluß an BS2000 7-500- und BM-Rechner
- Anschluß an das lokale Netz LAN Ethernet mit CSMA/CD (Carrier Sense Multiple Access/Collision Detection)
- Fest eingebautes Magnetbandkassettenaufwerk als schnelles Lademedium und zur Datensicherung; zusätzliches Magnetband-Laufwerk für Offline-Datenträgeraustausch und Datensicherung
- Teletex-Anschluß (Tbx) mit Zugang zu Telex (Tx)
- Bildschirmtext (Btx)
- Mehrplatzgrafik

Eine weitere Stärke des MX300 ist sein Betriebssystem SINIX, das UNIX von Siemens, mit einer von Siemens entwickelten Bedienoberfläche. UNIX ist ein leistungsfähiges und portables Betriebssystem, das weltweit akzeptiert wird.

Mit SINIX wird auch die Mehrplatzgrafik mit Fenstertechnik unter COLLAGE unterstützt.

Der MX300 wird eingesetzt

- im DDP (Distributed Data Processing)-Markt, mit geografisch verteilten und lokalen Datennetzen,
- im Projektgeschäft, mit individuellen Lösungen und
- als Stand-alone-System, mit fertigen Siemens-Anwenderlösungen.

NSC-Modelle

Die Leistungsmerkmale der SINIX-Mehrplatzsysteme innerhalb der NSC-Modelle sind gekennzeichnet durch:

- die Systemeinheit mit dem 32-bit-Prozessor NS32532/25 MHz
- den modularen Hauptspeicherausbau von 8-16 Mbyte
- den maximal 44 frei konfigurierbaren Schnittstellen, in Verbindung mit TACSI (Terminal Attachment Concept in SINIX).

Zwei Modelle stehen innerhalb der NSC-Prozessorlinie zur Verfügung:

- MX300-15 mit NS32532/25 MHz, maximal 12 Mbyte Hauptspeicher und 22 Schnittstellen, in Verbindung mit TACSI
- MX300-30 mit NS32532/25 MHz, maximal 16 Mbyte Hauptspeicher und 44 Schnittstellen, in Verbindung mit TACSI

Besondere Merkmale der NSC-Modelle

- Hauptspeicher in Ein-Megabit-Chip-Technologie
- Virtuelle Adressierung 4 Gbyte (= 10^9 Byte) Adreßraum
- Verarbeitungsprozessor NS 32532 mit 25 MHz Taktfrequenz
- Maximal 44 frei konfigurierbare serielle Ein-/Ausgabeschnittstellen bei Mitverwendung von TACSI
- Hochrüstung innerhalb der NSC-Prozessorlinie und zusätzlich Umrüstung auf die Intel-Prozessorlinie beim Kunden möglich.

Intel-Modelle

Die Steigerung der Leistungsmerkmale der SINIX-Mehrplatzsysteme wird innerhalb der neuen Intel-Modelle erreicht mit:

- der Systemeinheit, mit dem 32-bit-Prozessor Intel 80486 mit einem Grundtakt von 25 bzw. 33 MHz
- dem modularen Hauptspeicherausbau von 4-16 bzw. 16-64 Mbyte
- den maximal 54 frei konfigurierbaren Schnittstellen, in Verbindung mit TACSI

Drei Modelle stehen innerhalb der Intel-Prozessorlinie zur Verfügung:

- MX300-45 mit Intel 80486/25 MHz maximal 8 Mbyte Hauptspeicher und 6 Schnittstellen
- MX300-50 mit Intel 80486/25 MHz maximal 16 Mbyte Hauptspeicher und 44 Schnittstellen, in Verbindung mit TACSI
- MX300-60 mit Intel 80486/33 MHz maximal 64 Mbyte Hauptspeicher und 54 Schnittstellen, in Verbindung mit TACSI

Das System MX300 bietet optimale Wachstumsmöglichkeiten. So läßt sich z.B. jedes MX300-Modell stufenweise bis zum leistungsstärksten SINIX-Mehrplatzsystem MX300-60 direkt beim Kunden hochrüsten.

Besondere Merkmale der Intel-Modelle

- Hauptspeicher in Ein- bzw. Vier-Megabit-Chip-Technologie
- Virtuelle Adressierung 64 Tbyte (Terabyte = 10^{12} Byte) Adreßraum
- Verarbeitungsprozessor Intel 80486 mit 25 bzw. 33 MHz Taktfrequenz
- Maximal 54 frei konfigurierbare serielle Ein-/Ausgabeschnittstellen bei Mitverwendung von TACSI

Hardware

Gehäuse

Der MX300 ist integriert in einem 300 mm breiten Untertischmodell. Auf der Vorderseite dieses Gehäuses befinden sich das Bedienfeld, das Diskettenlaufwerk (bei MX300-45/-50/-60 zwei Diskettenlaufwerke möglich) und das Magnetbandkassettenlaufwerk (bei MX300-45 optional). Ein abschließbares Schiebefenster schützt das Bedienfeld und die Laufwerke vor unberechtigtem Zugriff.

Hauptspeicher

Der Hauptspeicher ist in 1 Megabit- oder 4 Megabit-Chip-Technik aufgebaut.

Bei den 1 Megabit-Bausteinen kommt die Parity-Sicherung zum Tragen. Das Hauptspeichermodul wird in Ausbaugrößen von 4 oder 8 Mbyte angeboten und kann durch HSP-Erweiterung um 2x4 Mbyte (Baby-Boards) auf 12 bzw. 16 Mbyte erhöht werden.

Beim Modell MX300-60 wird die neueste 4 Megabit-Chip-Technik verwendet. Hier kommt die Fehlerkorrektureinrichtung EDC (error-detection-correction) zum Einsatz, die 1 Bit-Fehler korrigiert und 2 Bit-Fehler erkennt. Das Hauptspeichermodul wird in Ausbaugrößen von 16 oder 32 Mbyte angeboten. Durch Erweiterung um 2x16 Mbyte (Baby-Boards) werden 48 bzw. 64 Mbyte erreicht.

Bussystem

Als MX300 Systembus wird das Standardbussystem MULTIBUS I^{*)} eingesetzt:

- 16 bit Datenbreite
- 2-3 Mbytes Datenrate

Abhängig vom Modell stehen max. 5 bzw. 12 Multibus I-Einbauplätze zur Verfügung.

Verarbeitungsprozessor

Bei den **NSC-Modellen MX300-15/-30** besteht der Verarbeitungsprozessor aus den Systembausteinen:

- Mikroprozessor NS 32532, 25 MHz
- Hauptspeicherverwaltung Memory Management Unit (MMU) on chip
- Gleitpunktprozessor Floating Point Unit (FPU) NS 32381/25 MHz

Der Mikroprozessor bietet:

- Interne Verarbeitungsbreite 32 bit
- Dynamische Anpassung der Busbreite auf 16 oder 32 bit, je nachdem ob über System- oder Speicherbus zugegriffen wird
- 4 Gbyte virtuellen Adreßraum bei 4 Kbyte Seitengröße
- Universalregister 8x 32 bit
- Grundtakt 25 MHz
- Prozessorleistung 8,0 MIPS

Bei den **Intel-Modellen MX300-45/-50/-60** kommt der Mikroprozessor Intel 80486 zum Einsatz. Er besteht aus den Systembausteinen:

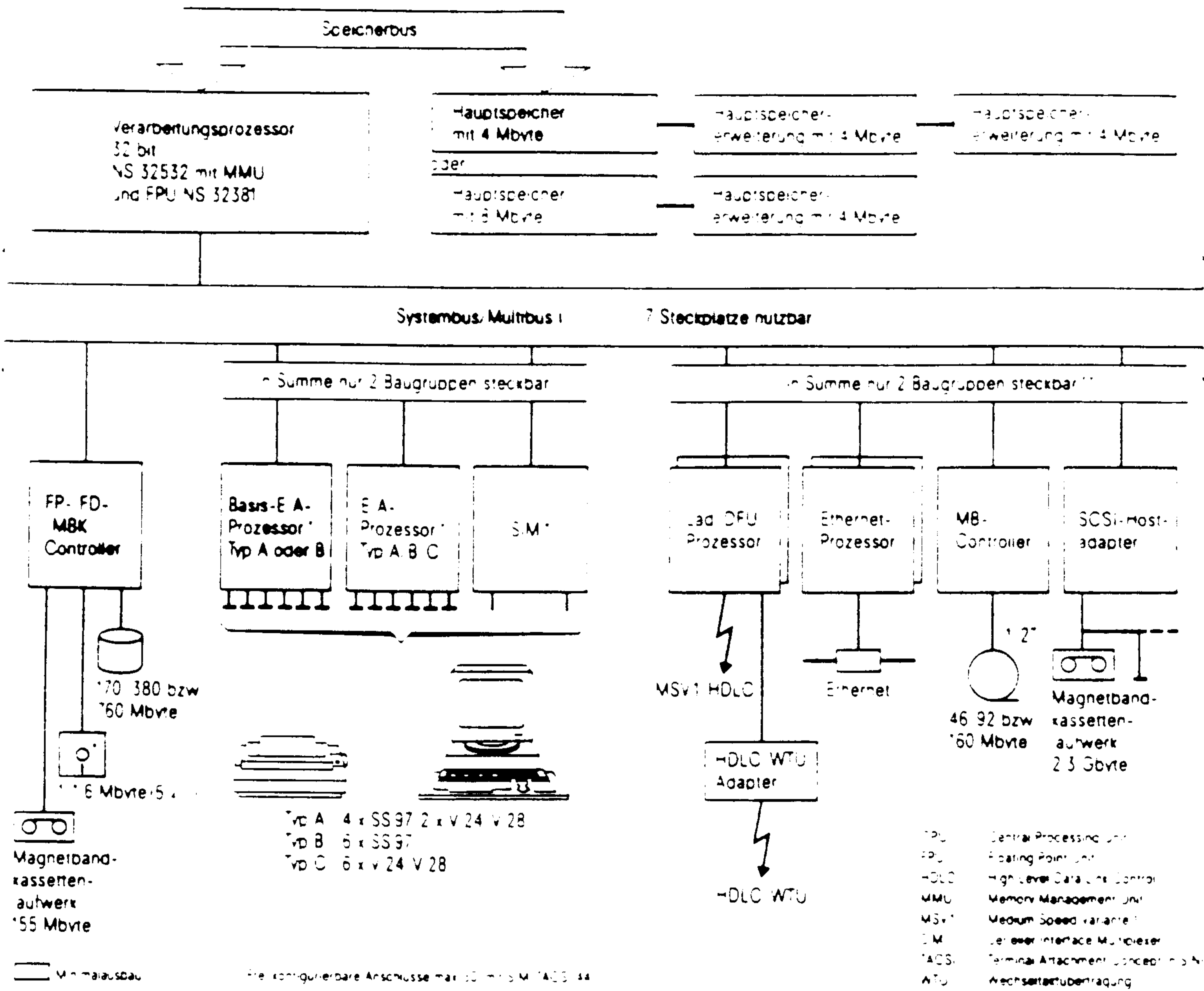
- Mikroprozessor 80486, 25 MHz bei MX300-45/-50
- Mikroprozessor 80486, 33 MHz bei MX300-60
- Hauptspeicherverwaltung Memory Management Unit (MMU) on chip
- Gleitpunktprozessor Floating Point Unit (FPU) on chip

Der Mikroprozessor bietet:

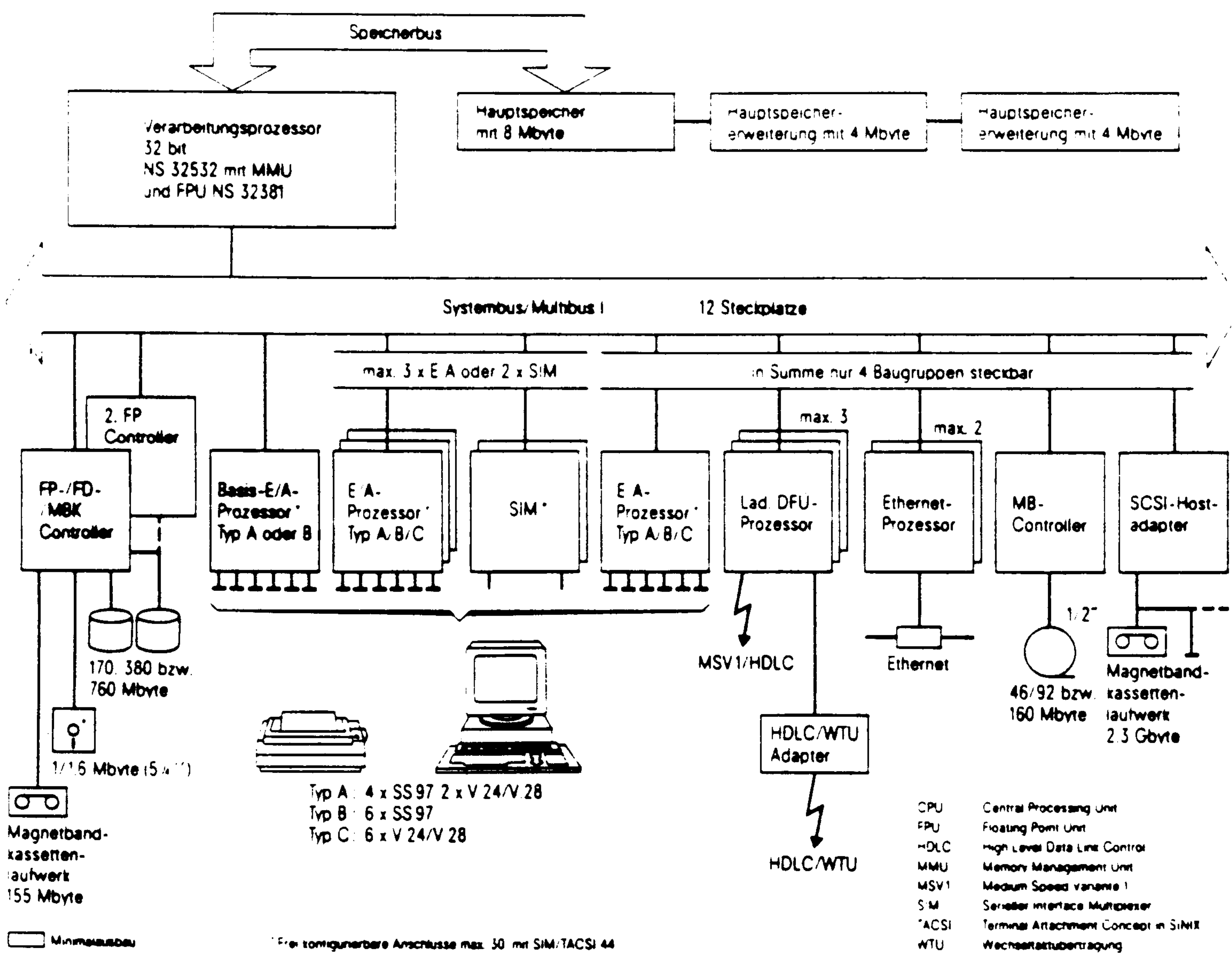
- interne Verarbeitungsbreite 32 bit
- Dynamische Anpassung der Busbreite auf 16 oder 32 bit, je nachdem ob über System- oder Speicherbus zugegriffen wird
- 64 Tbyte virtuellen Adreßraum bei einer logischen Segmentgröße von max. 4 Gbyte und einer linearen Seitengröße von 4 Kbyte
- Grundtakt 25 MHz bzw. 33 MHz
- Prozessorleistung 15 MIPS bei 25 MHz und 20 MIPS bei 33 MHz.

Die SiNIX-Mehrplatzsysteme der NSC-Linie und Intel-Linie gewährleisten innerhalb der Prozessorfamilien Objektcode-Kompatibilität und zueinander Sourcecode-Kompatibilität.

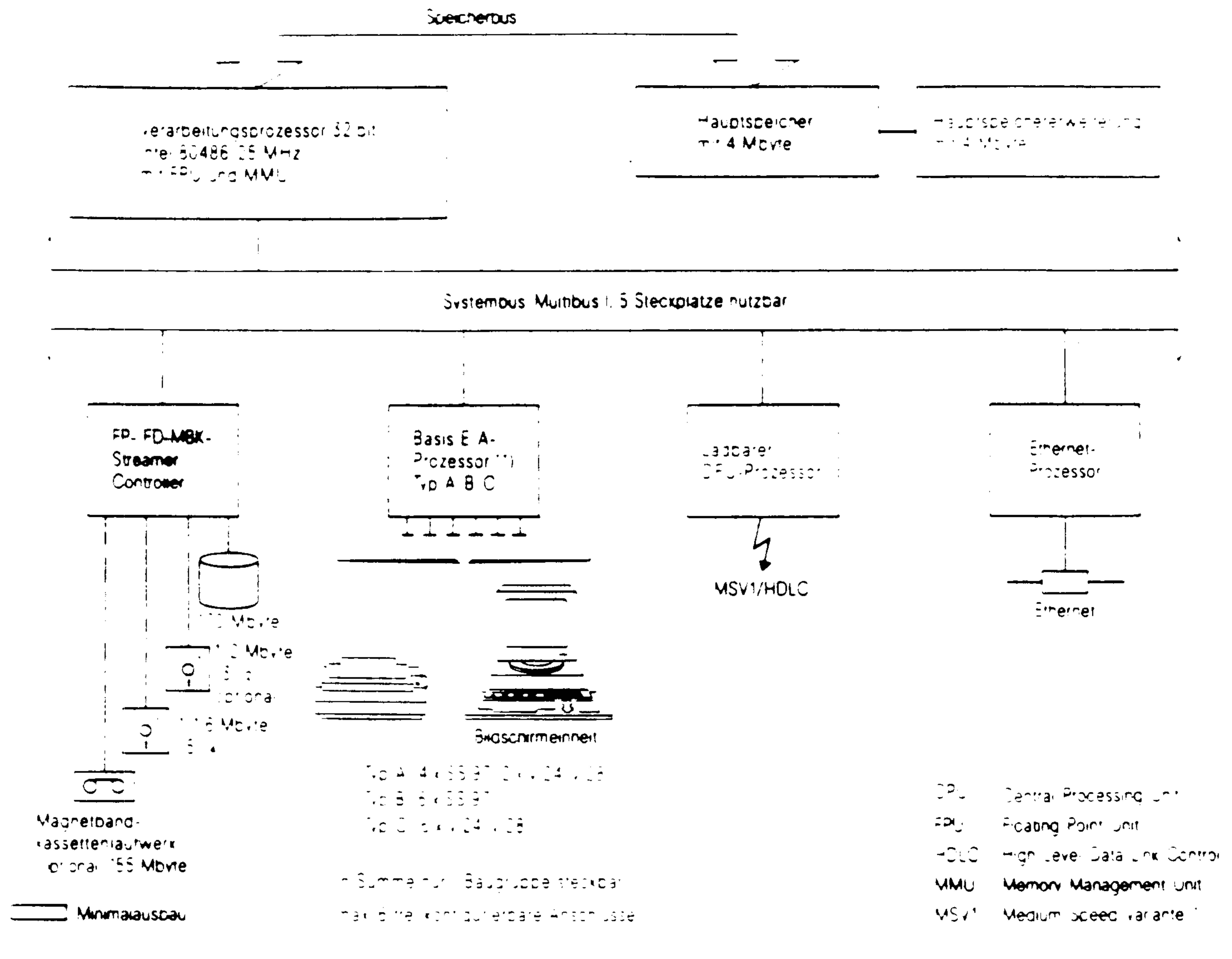
^{*)} Trademark of INTEL Corporation



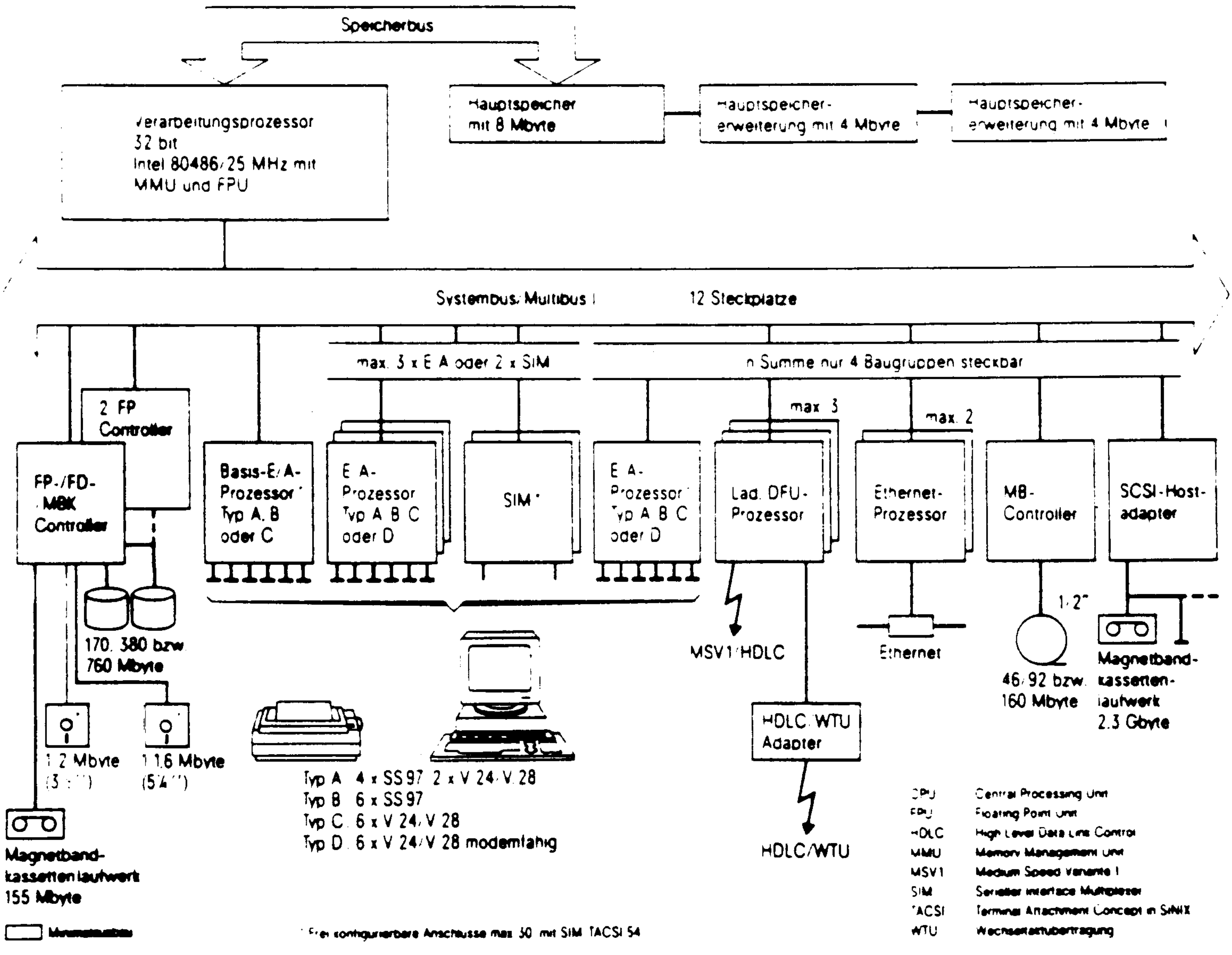
Konfigurationsbild MX 300-15



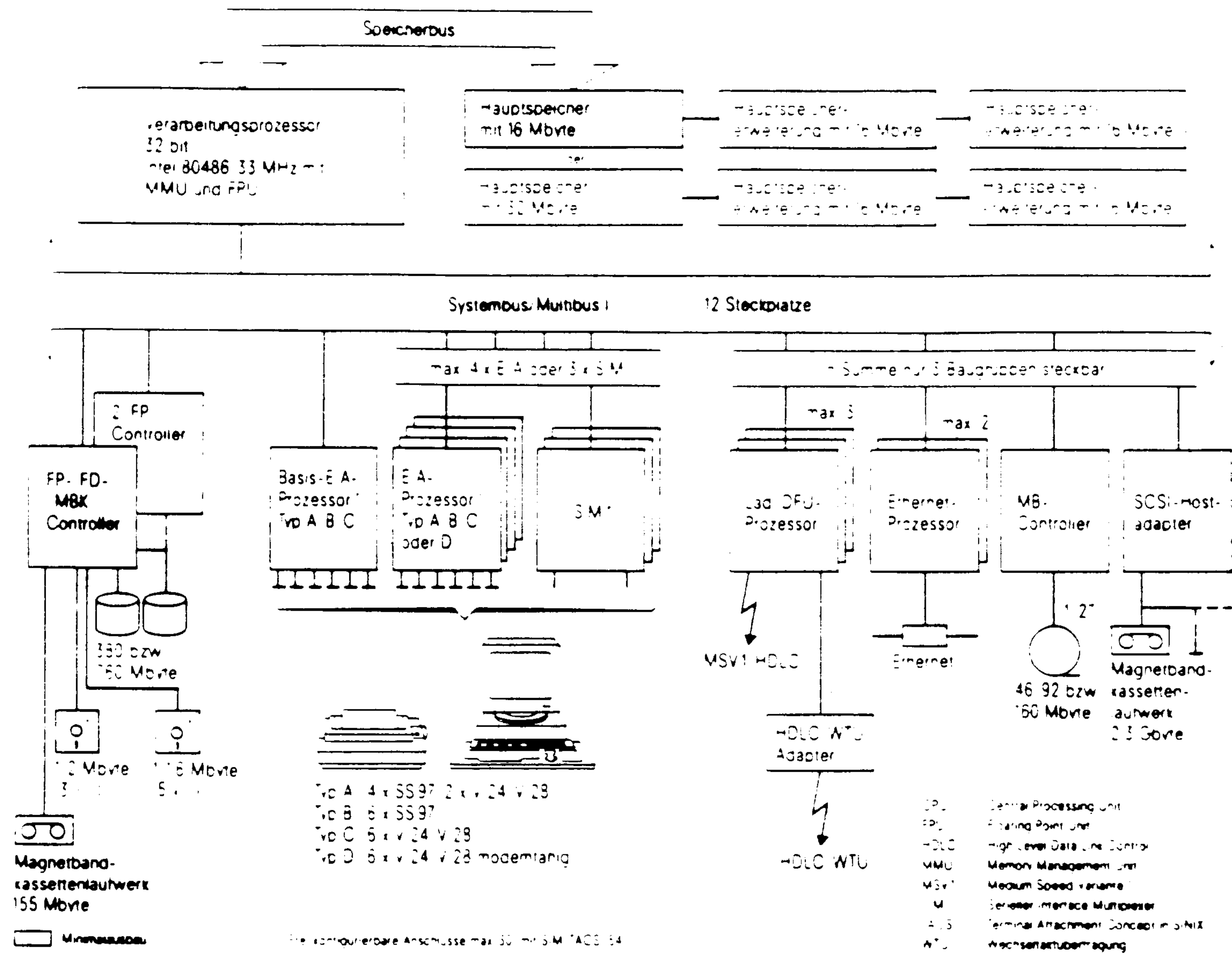
Konfigurationsbild MX 300-30



Konfigurationsbild MX 300-45



Konfigurationsbild MX 300-50



Konfigurationsskizze MX300-60

Nahperipherie

Standardmäßig enthalten alle MX300-Modelle:

- Diskettenlaufwerk (bimodal) 5 1/4" slim-line 1.0/1.6 Mbyte (unformatiert)
- Diskettenlaufwerk (bimodal) 3 1/2" slim-line 1.0/2.0 Mbyte unformatiert. Standardlaufwerk der Intel-Modelle.
- Magnetband-Kassettenlaufwerk 1", Formfaktor 5 1/4", Kapazität 155 Mbyte bei einer Kassettenbandlänge von 600ft. (beim MX300-45 optional).

Die Laufwerke sind bedienerfreundlich neben dem Bedienfeld untergebracht.

Zum Offline-Datenträgeraustausch (UNIX-tar-format und BS2000-Format) und als Backup-System mit logischer und physikalischer Sicherung kann auf Wunsch die

- Magnetbandeinheit (1") mit einer Bruttokapazität von 46 Mbyte bei 1600 bpi und von 92 Mbyte bei 3200 bpi oder die Magnetbandeinheit mit 160 Mbyte bei 6250 bpi eingesetzt werden.

Die Magnetbandeinheit befindet sich in einem Aufsichtsgenäuse und verfügt über eine Spulengröße bis 10.5" und automatische Bandenfädelung.

Zusätzlich bzw. alternativ zur Magnetbandeinheit steht für große Datenmengen zum Offline-Datenträgeraustausch und als Backup-System (logische physikalische Sicherung) auch ein

- Magnetband-Kassettenlaufwerk in Video-8-Technik mit einer Bruttokapazität von 2.3 Gbyte

zur Verfügung. Das Magnetband-Kassettenlaufwerk befindet sich in einem Aufsichtsgenäuse und wird über die SCSI-Schnittstelle mit dem MX300 verbunden.

Festplattenspeicher

In das Untertischgehäuse des MX300-45 ist nur ein Festplattenspeicher-Laufwerk mit 170 Mbyte Kapazität optional eingebaut.

In das Untertischgehäuse der übrigen MX300-Modelle können maximal 2 Festplattenspeicher-Laufwerke integriert werden. Drei Laufwerkstypen mit 170, 380 oder 760 Mbyte Kapazität (brutto) stehen zur Auswahl. Die Laufwerkstypen unterschiedlicher Kapazität können auch gleichzeitig eingesetzt werden.

Das zweite Festplatten-Laufwerk kann entweder über den Controller des ersten Festplatten-Laufwerks oder über einen zusätzlichen Festplatten-Controller betrieben werden.

Drucker

Als zentrale Drucker stehen zur Wahl:

- Banddrucker 9047
max. 600 Zeilen/min., nationale ASCII-Druckbander
- Seitendrucker 9025
elektrooptisch, 18 Seiten/min., ladbare Zeichensätze, Schonschriftqualität

Datenübertragung

Mit maximal drei ladbaren Dfu-Prozessoren werden folgende Kommunikationsmöglichkeiten geboten:

- Stationskopplung zum BS2000
- Zugriff auf BS2000-Datenbanken und Dialoganwendung im BS2000
- File Transfer zwischen BS2000 und SINIX Programm-Programm-Kommunikation
- Rechnerkopplung zu BS2000/TRANSDATA (NEA- und ISO-Protokolle) und zu OSI-Partnern
- SNA-Anbindung (SDLC)

Ethernet-Anschluß

Mit dem ladbaren, autonomen Ethernet-Prozessor kann der MX300 an lokale Netze auf Ethernet-Basis mit dem Zugriffsverfahren CSMA/CD (Carrier-Sense-Multiple-Access/Collision Detection) angeschlossen werden. Das ladbare Kommunikationstransportsystem läßt die Wahl zu zwischen TCP/IP (Transmission-Control-Protocol, Internet Protocol) und den OSI-Protokollen.

Teletex-Anschluß (Ttx)

Der Teletex-Anschluß ermöglicht den Zugang zum weltweiten und herstellerübergreifenden Teletex-/Telex-Netz.

Bildschirmtext (Btx)

Ein Btx-Decoder eröffnet den Zugang zum öffentlichen Btx-Dienst der DBP. Dem Benutzer steht ein zusätzlicher Btx-Monitor zur vollständigen Darstellung der Btx-Seiten, sowohl als „Einschirm-“ als auch mit einer „Zweischirm-Lösung“ zur Verfügung. Die Bedienung erfolgt über den SINIX-Bildschirmarbeitsplatz.

TELESERVICE-Zentrale

Mit einem Zusatz läßt sich der MX300 zur TELESERVICE-Zentrale erweitern, um dezentrale SINIX-Systeme per Ferndiagnose betreuen zu können. Die zu betreuenden SINIX-Systeme müssen einen entsprechenden TELESERVICE-Zusatz enthalten. Der Anschluß erfolgt über Modem (duplex asynchron) und bestehende Telefonnetze.

TELESERVICE-Zusatz

Der TELESERVICE-Zusatz ermöglicht den Anschluß des MX300 an ein Service-Zentrum zur Nutzung der Ferndiagnose und Fehlerkorrektur. Der Anschluß erfolgt über Modem (duplex asynchron) und bestehende Telefonnetze.

Unterbrechungsfreie Stromversorgung (USV)

Bei Netzausfall sorgt sie für die erforderliche Zeit, um ein ordnungsgemäßes Abschalten (shut-down) des MX300 sicherzustellen. Dabei ist die Auswahl zwischen sofortigem oder einem um 20 Sekunden verzögerten Abschalten möglich.

Arbeitsplatzkomponenten

An das Mehrplatzsystem MX 300 können alphanumerische und grafische lokale SINIX-Bildschirmarbeitsplätze angeschlossen werden. In Verbindung mit TACSI sind modellabhängig maximal 44 frei konfigurierbare serielle Schnittstellen für die NSC-Linie und maximal 64 für die Intel-Linie vorhanden.

Ein Arbeitsplatz besteht aus:

- Bildschirmeinheit 97801 mit 14" Bildschirm, Alphamodus, schwarz-weiß, Positivdarstellung mit nationalen Tastaturen oder
- Bildschirmeinheit 97808 mit 15" Bildschirm, Alpha- oder Grafikmodus wahlbar, schwarz-weiß, Positivdarstellung mit nationalen Tastaturen und dem Bedienelement Maus
- Arbeitsplatzdrucker
 - Drucker 9001, Matrixdrucker mit Tinte (Zeichendrucker)
 - Drucker 9011, Matrixdrucker mit Nadel (Zeichendrucker)
 - Drucker 9012, Matrixdrucker mit Tinte (Zeichendrucker)
 - Drucker 9013, Matrixdrucker mit Nadeldruck
 - Drucker 9021, 9022, Elektrooptischer Drucker mit Schonschriftqualität

Alle Arbeitsplatzkomponenten sind mit der Schnittstelle SS97/V.11 ausgestattet und können im Direktanschluß bis 60 m angeschlossen werden, mit Ausnahme des Druckers 9022, der über V.24 angeschlossen wird. Die Datenübertragungsgeschwindigkeit beträgt maximal 38,4 kbit/s.

Terminal Attachment Concept in SINIX (TACSI)

Die SINIX-Arbeitsplätze können durch TACSI über ungeschirmte Zweidrahtleitungen bis zu 2000 m entfernt betrieben werden. TACSI besteht aus dem Seriellen-Inhouse-Multiplexer (SIM) und dem Terminal-Anschluß-Konzentrator (TAK). Maximal vier TAK lassen sich an einem SIM anschließen; pro TAK stehen vier SS97 V.24-Schnittstellen zur Verfügung. Somit können bis 16 Endgeräte über einen TACSI-Anschluß betrieben werden. TACSI erlaubt z.B. beim MX 300-60 die Erhöhung der Anzahl serieller Terminalschnittstellen von 30 auf 54. Die Anzahl der Terminals ist dabei abhängig von der Anwendung und dem Terminaltyp.

Software

Das Betriebssystem SINIX besteht aus:

- Benutzeroberfläche
- Systemprogrammen
- Kommandoabfolgen
- Systemkern

Über eine hierarchische Menüs greift der Benutzer auf Systemfunktionen und Anwendungen zu.

- Online-Betrieb an BS2000- oder BM-Hosts mit gleichem Verhalten wie dem einer Datensichtstation 9750 oder 3270
- Direkter Datenaustausch z.B. von dezentral ertägten und überprüften Daten mit der zentralen DVA über File-Transfer
- Bedienungsführung am Bildschirm (Menu- bzw. Fenstertechnik)
- Datensicherung über Diskette, Magnetbandkassette oder Magnetbandgerät
- Freie Programmierung in mehreren Sprachen
Weitere Informationen können Sie den Software-Produktblättern entnehmen.

Eine Vielzahl von betriebssystemnaher Software und Anwendersoftware steht zur Verfügung. Über die verfügbare Software informieren Sie den „Software-Produkte-Katalog SINIX“ und die „Softwarecourse SINIX“.

Ausbaumöglichkeiten des MX 300

Modell	MX 300-15	MX 300-30
Verarbeitungsprozessor	1xS 32532/25 MHz	1xS 32532/25 MHz
Hauptspeicher-Ausbau (Mbyte)	4-12	8-16
Multibus Einbauplatze	7 nutzbar	12
E/A-Prozessor	1-2	1-5
Serieller Interface Multiplexer (SIM)	1	2
Frei konfigurierbare serielle Schnittstellen (mit TACSII)	6-12 (22)	6-30 (44)
Disketten-Laufwerk 3 1/2"	-	-
Disketten-Laufwerk 5 1/4"	1	1
Magnetbandkassetten-Laufwerk	1	1
Festplatten-Laufwerk (70/380/760 Mbyte)	1	2
DFU-Prozessor	2 ¹⁾	3 ²⁾
LAN (Ethernet)-Prozessor	2 ³⁾	2 ³⁾
Plattencontroller	1	2
Magnetbandgerät 1/2"	1	1
Magnetband-Kassettenlaufwerk 2.3 Gbyte	1	1

Modell	MX 300-45	MX 300-50	MX 300-60
Verarbeitungsprozessor	intel 80486/25 MHz	intel 80486/25 MHz	intel 80486/33 MHz
Hauptspeicher-Ausbau (Mbyte)	4-8	8-16	16-64
Multibus Einbauplatze	5 nutzbar	12	12
E/A-Prozessor	1	1-5	1-5
Serieller Interface Multiplexer (SIM)	-	2	3
Frei konfigurierbare serielle Schnittstellen (mit TACSII)	6	6-30 (44)	6-30 (54)
Disketten-Laufwerk 3 1/2"	1	1	1
Disketten-Laufwerk 5 1/4"	1 (optional)	1 (optional)	1 (optional)
Magnetbandkassetten-Laufwerk	1 (optional)	1	1
Festplatten-Laufwerk (70/380/760 Mbyte)	1 (170)	2	2 (380/760)
DFU-Prozessor	1 ¹⁾	3 ²⁾	3 ⁴⁾
LAN (Ethernet)-Prozessor	1 ³⁾	2 ³⁾	2 ⁴⁾
Plattencontroller	1	2	2
Magnetbandgerät 1/2"	-	1	1
Magnetband-Kassettenlaufwerk 2.3 Gbyte	-	1	1

max. 1 ¹⁾ in Summe max. 4

²⁾ in Summe max. 2

³⁾ in Summe max. 3

Technische Daten

Systemeinheit	MX 300-15/-30	MX 300-45/-50	MX 300-60
Mikroprozessor	NS 32532/25 MHz	Intel 80486 25 MHz	Intel 80486 33 MHz
Verarbeitungsbreite (bit)	32	32	32
Adressbusbreite (bit)	24	24	24
Speicherbusbreite (bit)	32	32	32
Adressierung	4 Gbyte virtuell	64 Tbyte virtuell	64 Tbyte virtuell
Hauptspeichererweiterung (MMU)	on chip	on chip	on chip
Gleitkomprozessor (FPU)	NS 32381/25 MHz	on chip	on chip
Systembus	INTEL Multibus I	INTEL Multibus I	INTEL Multibus I
Hauptspeicher (Mbyte)	4 bis 12 (MX 300-15) 8 bis 16 (MX 300-30)	4 bis 8 (MX 300-45) 8 bis 16 (MX 300-50)	16 bis 64
Echtzeituhr	batteriegepuffert	batteriegepuffert	batteriegepuffert
Performance-Werte			
Prozessorleistung in MiPS (1 MiPS \approx CPU-Leistung VAX 11/780 ¹)	8,0	15-20 ²	20-27 ²
Dhrystone KDhry/sek (Dhrystone V1.0)	9,3	25-35 ³ (Dhrystone V1.1)	23-47 ³
Laufwerke			
	im Grundausbau enthalten:		optional:
	3 1/2"-Diskette	5 1/4"-Diskette	1/4"-MB-Kassette²⁾
Kapazität (Mbyte, brutto)	1,0/2,0	1,0/1,6	1/4"-Magnetband 46 (1600 bpi) 92 (3200 bpi) 160 (6250 bpi) ³
(Mbyte, netto)	0,72/1,44	0,64/1,2	Video-8 MB-Kassette 2300
Spuren/Oberfläche	80/2	80/2	-
Du-Rate (Mbyte/s)	0,031/0,062	0,031	0,09
Köpfe (R/W + Servo)	2	2	0,040/0,160/ 0,312 ³ /0,625 ³
Drehzahl (min ⁻¹)	300	300	1-9 Spur-Kopf
Latenzzeit (ms)	100	100	-
Positionierzeit			-
min./mittl./max. (ms)	3/79/255	3/79/255	-
Bandgeschwindigkeit (m/s)	-	-	2,29
Aufzeichnung	-	-	0,635/2,54 1,27
			18 Spur (serpentine)
			9 Spur (parallel)
			helical scan recording
Laufwerke			
	notwendige Komponenten: 5 1/4"-Festplatten		
Kapazität (Mbyte, brutto)	170	380	760
(Mbyte, netto)	146	303	631
Spuren/Oberfläche	1024	1218	1626
Du-Rate (Mbyte/s)	1,25	1,25	1,37
Köpfe (R/W + Servo)	8+1	15+1	15+1
Drehzahl (min ⁻¹)	3600	3600	3600
Latenzzeit (ms)	8,3	8,3	8,3
Positionierzeit			
min./mittl./max. (ms)	6/28/62	4/18/40	4/16/35
Umgebungsbedingungen			
	MX 300	Magnetbandgerät	MBK-LW-Video-8
Netzspannung (V)	110-120 \pm 10% 220-240 +6%/-10%	110-120 \pm 15% 220-240 \pm 15%	120-240 +10%-20% -
Netzfrequenz (Hz)	50-60 \pm 1%	50-60 \pm 5%	50-60 \pm 5%
Leistungsaufnahme (kVA)	0,5 (MX 300-15/-45) 1,0 (MX 300-30/-50/-60)	0,35	0,052
Wärmeabgabe (kJ/h) geschätzt	1400 (MX 300-15/-45) 2300 (MX 300-30/-50/-60)	1000	60
Klimaklasse (allg. Büroklima)	2	2	2
Geräuschpegel dB(A)	<50	-	-

¹ VAX is a trademark of Digital Equipment Corporation

² Bei MX300-45 optional

³ Wert gilt für GCR-Magnetband

⁴ Maximalwert nach Herstellerangaben Intel

Technische Daten Fortsetzung

Maße und Gewicht	MX 300	Magnetbandgerät	MBK-LW-Video-8
H x B x T (mm)	Untertischgehäuse 595 x 300 x 690	Auftischgehäuse 260 x 505 x 690	Auftischgehäuse 127 x 229 x 335
Gewicht (kg)	35 (MX 300-15) -45 45 (MX 300-30) -50 -60, ...	50	6,8
SINIX-Bildschirmarbeitsplätze	alphanumerisch	grafisch	
Bildschirm-Typ	97801 (14")	97808 (15")	
Bildschirmdiagonale (mm)	356	381	
Bildschirmfarbe (Positivdarstellung)	schwarz/weiß	schwarz/weiß	
Bildwiederholfrequenz (Hz)	> 70	> 70	
Zeichen pro Zeile	80 bei 25 Zeilen	80 bei 25 Zeilen (Alpha)	
Zeichendarstellung 7x9			
in Punktmatrix	9x16	9x16	
Bildschirmauflösung (Pixel)	-	720x540 (Grafik) 720x400 (Alpha)	
Leuchtdichte (cd/m ²)	35	≤ 85	
Schnittstelle zum System	SS97/V.11	SS97/V.11	
Übertragungsgeschw. (kbit/s)	38.4	38.4	
Betriebsart	Alphamodus	Alpha- und Grafikmodus wählbar	
Leitungslänge max. (m)	60	60	
mit AFP-Zusatz max. (m)	2000	2000	
Umgebungsbedingungen	Bildschirmeinheit 14"	Bildschirmeinheit 15"	
Netzspannung (V)	110-120 +6%/-10%	110-120 +6%/-10%	
	220-240 +6%/-10%	220-240 +6%/-10%	
Netzfrequenz (Hz)	50-60	50-60	
Leistungsaufnahme (kVA)	0,12	0,16	
Wärmeabgabe (kJ/h)	220	470	
Klimaklasse (allg. Büroklima)	2	2	
Maße und Gewicht	Bildschirmeinheit 14"	Bildschirmeinheit 15"	
H x B x T (mm)	410 x 370 x 400	380 x 370 x 350	
Gewicht (kg) ca.	15	14	
	Tastatur	Steuereinheit	
	40 x 470 x 200	109 x 375 x 384	
	2	8	
		Tastatur und Maus	
		40 x 470 x 200	
		2	
Zulassungen und Sicherheitsprüfungen	allgemeine Betriebsgenehmigung gemäß Postverordnung 1046/84		
ETZ-Zulassung	allgemeine Betriebsgenehmigung gemäß Postverordnung 1046/84		
Kommunikationszulassung	ZZF-Nr. A303205W		
Sicherheitsprüfung	VDE 0805, UL 478		
Emissionsprüfung	FCC Class A		

✓

✓

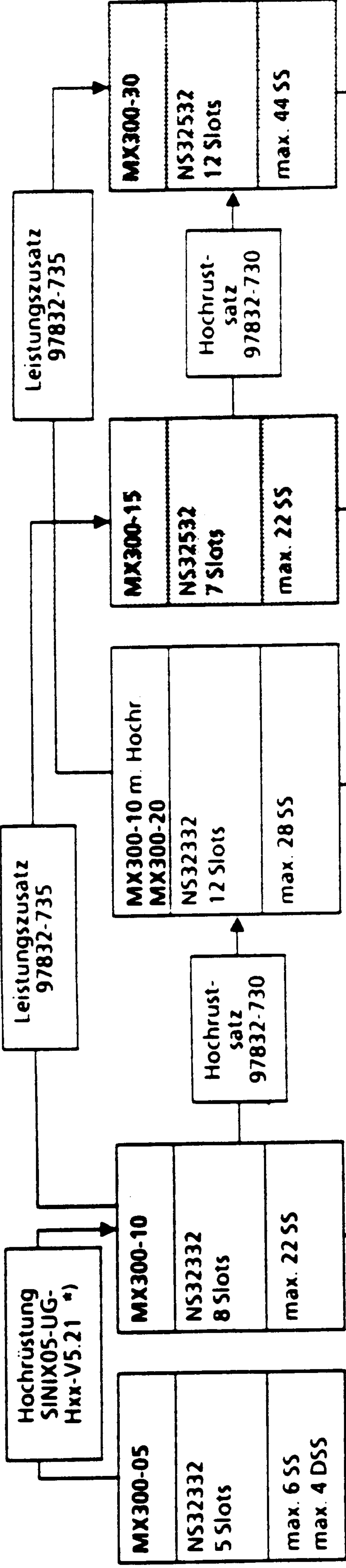
✓

✓

VARIANTE DI MODELLI

Hochrüstung der MX300-Modelle

NSC-Modelle



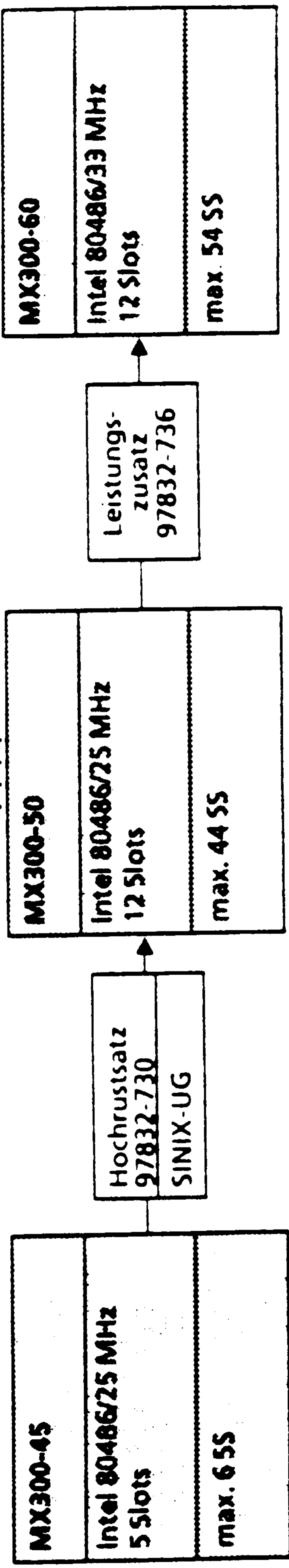
Min. 8 Mbyte HSP
Nachrüstung
97832-731

Nachrüstung
97832-732

Min. 8 Mbyte HSP
Nachrüstung
97832-731

Nachrüstung
97832-731

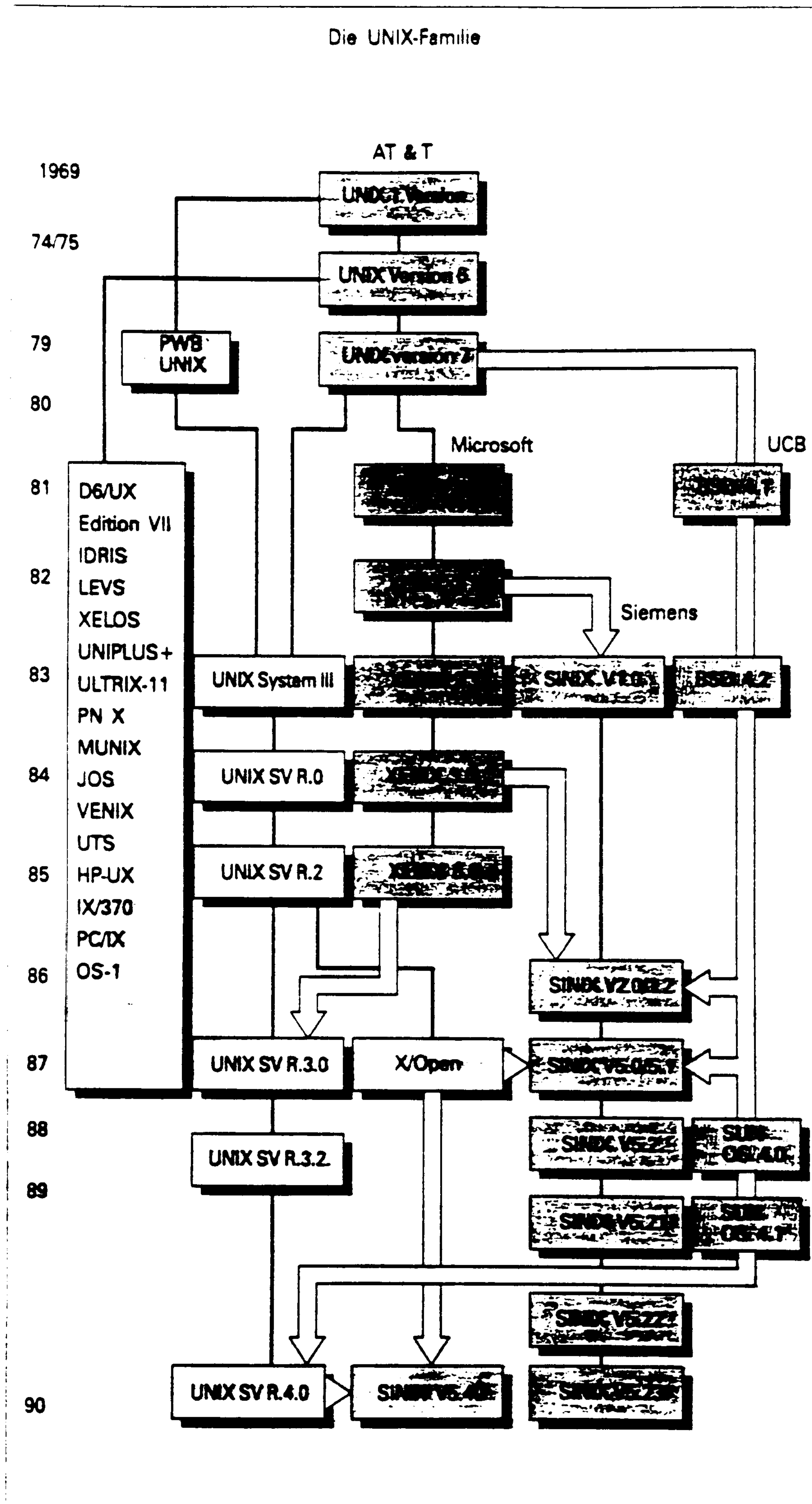
MODELLO INTEL Intel-Modelle



☐ "Aktuelle Modelle"

*) SW-Update von SINIX V5 21 auf V5 22 ist gesondert zu bestellen!

UNIX - Familie



Softwareprodukt

SINIX Version 5.40

Betriebssystem für SINIX-Systeme MX300-45*/-50/-60

Produktblatt

Ausgabe 7/90

Charakterisierung des Produktes

SINIX ist das UNIX™ von Siemens.

UNIX ist ein ausgereiftes und universell einsetzbares Betriebssystem, das 1969 erstmals in den Bell Laboratories von AT&T entwickelt wurde und das sich als Weltstandard für Betriebssysteme im Mehrplatz-computerbereich durchgesetzt hat.

UNIX hat eine Reihe von - im Vergleich zu traditionellen Betriebssystemen für Büro- und Minicomputer - wichtigen Eigenschaften:

- **Parallelverarbeitung (Multi-Tasking-Fähigkeit)**

Ein Benutzer kann mehrere Aufträge (jobs), die für den Rechner in Aufgaben (tasks) gegliedert sein können, gleichzeitig ausführen lassen.

- **Mehrbenutzer (Multi-User)-Betrieb**

Mehrere Benutzer können über unterschiedliche Terminals gleichzeitig mit dem System arbeiten. Durch das gemeinsam genutzte Datenhaltungssystem wird die Datenkonsistenz innerhalb einer Organisationseinheit gewährleistet.

- **Hierarchisch strukturiertes Dateisystem**

UNIX besitzt ein geräteunabhängiges Dateisystem, das die Organisation und die Verwaltung hierarchischer Strukturen auf einfache Weise unterstützt.

- **Programmketten (Pipelining)**

Filter und Pipes sind Mechanismen zur Bildung von Programmketten für komplexe Funktionen aus einfachen Bausteinen. In einer Pipe dient die Ausgabe eines Filters als Eingabe für den nächsten Filter.

- **Kommunikation im LAN**

Das als Industrie-Standard akzeptierte Transportprotokoll TCP/IP ist Bestandteil des Betriebssystems und dient als Transportsystem für höhere Kommunikationsdienste. Mit DFS (NFS) kann ein netzweit verteiltes Dateisystem aufgebaut werden.

• Das HW-/SW-Paket MX300 beinhaltet nur das Laufzeitsystem von SINIX V5.40. In diesem sind nicht enthalten: CES, „Spiegelplatten“ und die Anschlußfähigkeit bestimmter Peripheriegeräte.

- **Befehls- und Dienstprogramm-Umfang**

Seine Mächtigkeit erhält UNIX durch die große Anzahl an Kommandos und Dienstprogrammen, aus denen sich maßgeschneiderte Anwendungen zusammenstellen lassen.

SINIX V5.40 basiert auf AT&T's aktuellem UNIX System V Release 4.0, das die wichtigsten UNIX-Linien, internationale und De-facto-Standards sowie State-of-the-Art-Technologie in sich vereinigt.

SINIX weist darüber hinaus weitere wichtige Merkmale auf:

- erweiterte X/Open-Konformität,
- ein flexibles Kommunikationskonzept,
- ein benutzerfreundliches, ergonomisches Bediensystem mit Fenstertechnik,
- Bedienerführung und Dokumentation in Deutsch und Englisch sowie die Option auf weitere Sprachen,
- erweiterte Spool-Funktionen,
- Basisfunktionen ("Spiegelplatte"), um eine höhere Systemverfügbarkeit zu erreichen,
- integriertes C Entwicklungssystem (CES).
- Programmauswahl/-vielfalt durch Zusammenarbeit mit qualifizierten Softwarehäusern (comPartnern) und
- weite Verbreitung (Nr. 1 in Europa bei UNIX-Mehrplatzsystemen).

Siemens engagiert sich als Gründungsmitglied der X/Open-Group - einer internationalen Vereinigung führender Hard- und Software-Hersteller zur Standardisierung von UNIX - von Anfang an sehr stark für diese Zielsetzung und unterstützt aktiv die Arbeiten von X/Open.

Die Ergebnisse von X/Open werden im sogenannten Portability-Guide (neueste Ausgabe: XPG3) veröffentlicht. Dazu gehören u.a.:

- Festlegungen zur Betriebssystem-Schnittstelle, basierend auf den POSIX Festlegungen IEEE 1003.1),
- Window Management,
- Vereinbarungen über die Datenverwaltung (Zugriffsmethode ISAM, Abfragesprache für relationale Datenbanken SQL),
- wichtige Programmiersprachen (C, COBOL, FORTRAN, PASCAL),
- einheitliche Datenträgeraustauschformate (Disketten, Magnetband) und
- Internationalisierung: nationale Zeichensätze (8-bit-Code), Message-Kataloge, nationale Konventionen.

SINIX V5.40 ist POSIX- und X/Open-konform (BASE XPG3). In Verbindung mit systemnahen SW-Komponenten wird PLUS XPG3-Konformität erreicht (Verifizierung in Vorbereitung).

Die in XPG3 getroffenen Festlegungen zur 8-bit-Transparenz und von 8-bit-Zeichensätzen (ISO 8859) sind auch in den Utilities weitgehend realisiert. Mit der Verfügbarkeit von 8-bit-Terminals und 8-bit-Druckern sind die Voraussetzungen für eine durchgängige Verwendung nationaler Zeichensätze geschaffen.

Beschreibung der Funktionen

Der SINIX-Betriebssystem-Kern umfaßt alle Funktionen für den Ablauf von System- und Anwenderprogrammen. Die Prozeßverwaltung mit virtueller Adressierung, Realtime Unterstützung, Gerätetreiber für die Peripheriegeräte und Gleitpunktarithmetik sind ebenso Bestandteil des Betriebssystemkerns wie ein effizientes Dateisystem. Darüber hinaus enthält SINIX neben UNIX-üblichen und durch X/Open spezifizierten Utilities eine C-Entwicklungsumgebung sowie SINIX-spezifische Erweiterungen wie z.B. eine ergonomische Bedienoberfläche mit deutscher und englischer Bedienungsführung, ein leistungsfähiges Spool-System und die Möglichkeit Spiegelplatten zu führen.

Prozeßverwaltung

Für jedes ablaufende Programm wird ein Prozeß erzeugt, der selbst wieder Unterprozesse erzeugen und verwalten kann. Damit sind die Basismechanismen für Multi-Tasking- und Multi-User-Betrieb von SINIX gegeben.

Wird ein Programm mehrfach aufgerufen, so können die ausführenden Prozesse den invarianten Teil eines Programms (Programm-Code) gemeinsam benutzen, während der Datenbereich für jeden Prozeß separat angelegt wird.

Die Speicherverwaltung (Memory Management) bildet die virtuellen Adressen auf den realen Hauptspeicher ab. Übersteigt der Platzbedarf aller Prozesse den realen Hauptspeicher, so werden momentan nicht benötigte Speicherbereiche (Seiten oder Pages) auf Platte ausgelagert und später bei Bedarf wieder eingelagert (Demand Paging).

Die maximale Programmgröße ist durch den für das Paging verfügbaren Plattenspeicherbedarf begrenzt (abhängig von Plattenanzahl und -typ).

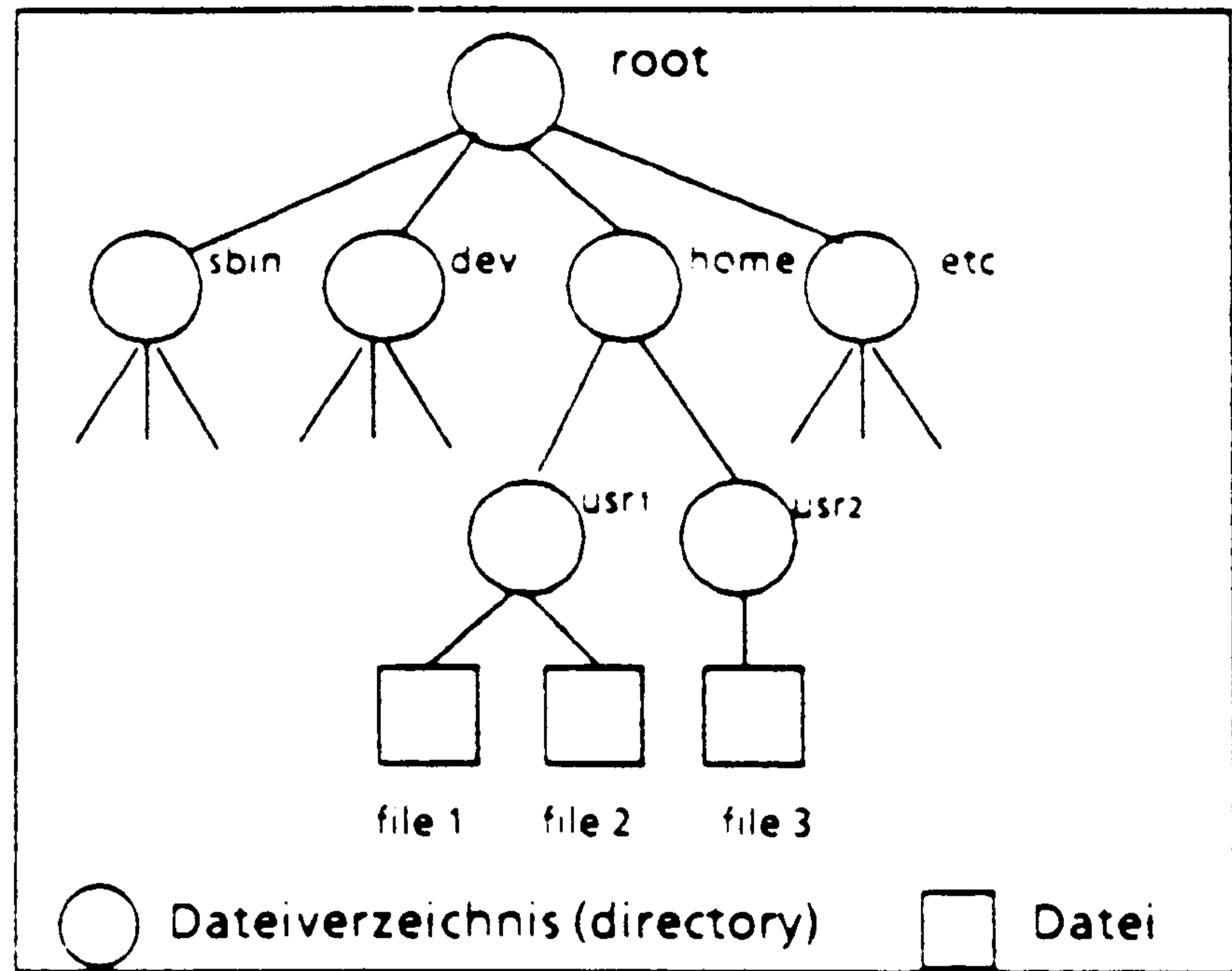
Aufgrund der virtuellen Adressierung sind die einzelnen Prozesse voreinander geschützt, wodurch wesentlich zur Betriebssicherheit des Systems beigetragen wird.

Dateisystem

Die Daten werden von einem einfachen und leistungsfähigen Dateisystem verwaltet, das sich u.a. auszeichnet durch:

- die baumartige Struktur des Dateisystems,
- die Zugriffskontrolle für Dateien und
- die Geräteunabhängigkeit.

SINIX-Dateisystem



Die Blätter des Baumes (im Bild mit einem Rechteck symbolisiert) sind Dateien im üblichen Sinn und enthalten Programme oder Daten. Die übrigen Knoten des Baumes - die Dateiverzeichnisse - sind zwar auch Dateien, aber mit einer ganz speziellen Bestimmung: In ihnen sind Informationen über andere Dateien verzeichnet. Durch diese Struktur können auch in großen Katalogen einzelne Dateien sehr rasch gefunden werden.

SINIX bietet eine Reihe funktionsstarker Kommandos zur Dateiverwaltung, wobei die hierarchische Ordnung des Dateisystems sinnvoll ausgenutzt wird. Zum Beispiel können mit einem einzigen Kommando alle Dateien eines Verzeichnisses und aller Unterverzeichnisse kopiert werden.

Zugriffskontrolle für Dateien:

Für jede Datei können differenziert Zugriffsberechtigungen vergeben werden.

Die Zugriffserlaubnis kann gelten für

- den Eigentümer,
- die Benutzergruppe des Eigentümers und
- alle anderen Benutzer.

Der Zugriff kann zugelassen werden

- zum Lesen,
- zum Schreiben und
- zur Ausführung (Programme, Prozeduren).

Geräteunabhängigkeit

Die Peripherie-Geräte werden vom Betriebssystem ebenfalls wie Dateien behandelt. Mit ihnen wird neben den normalen Dateien und den Dateiverzeichnissen eine dritte Klasse von Dateien eingeführt: die Gerätedateien. Sie ermöglichen große Flexibilität bei der Behandlung der Ein-/Ausgabe in den Anwendungen: Die Festlegung der Peripherie-Geräte muß nicht wie üblich bei der Programmierung, sondern kann situationsbezogen beim Ablauf des Programms erfolgen.

Realtime-Fähigkeit

Zur Realisierung von Echtzeitanforderungen enthält SINIX V5.40 zwei Erweiterungen:

1) Realtime Scheduling

Ein Prozeß-Scheduler legt fest, welches Programm als nächstes läuft und wie lange es läuft.

SINIX V5.40 unterstützt neben dem konventionellen (Timesharing) Scheduling eine neue Realtime-Scheduling-Strategie. Für jeden Prozeß kann zur Laufzeit festgelegt werden, unter welcher Scheduler-Strategie er ablaufen soll.

2) High Resolution Timer

Für den Timer gibt es eine sehr feine Granularität, um auch Ereignisse, die in sehr kurzen Intervallen auftreten, behandeln zu können.

Peripherie

In SINIX V5.40 werden folgende Hardware-Komponenten unterstützt:

- alphanumerischer Bildschirm 97801,
- Grafikbildschirm 97808,
- Drucker 9001, 9004, 9011, 9012, 9013, 9014 (über SS97),
- Drucker 9022, 9025, 9047 (über RS232),
- Festplattenlaufwerke 5 1/4" (170/380/760 Mbyte),
- Diskettenlaufwerke 3 1/2" und 5 1/4" (bis 2,0 Mbyte),
- MB-Kassettenlaufwerk 1/4" (155 Mbyte),
- MB-Kassettenlaufwerk Video 8 (bis 2,3 Gbyte),
- Magnetbandgerät 1/2" (bis 160 Mbyte),
- batteriegepufferte Realzeituhr (nicht flüchtig),
- ladbare(r) Kommunikationsprozessor(en),
- Chipkarten-Terminal,
- modemfähiger E/A-Prozessor (SEAAC),
- Terminal Anschlußkonzept (TACSI) und
- Unterbrechungsfreie Stromversorgung (USV).

Benutzerfreundliches Bediensystem mit Fenster-technik

Neben dem Experten-Modus (Shell) bietet SINIX V5.40 eine ergonomische, objekt-orientierte Bedienoberfläche mit Fenstertechnik, die sowohl auf grafischen als auch auf alphanumerischen Bildschirmen für eine einheitliche Systembedienung sorgt.

Erweiterte Spool-Funktionen

Das SINIX-Spool-System bietet zusätzliche Funktionen wie:

- Zustandsüberwachung der konfigurierten Drucker und der laufenden Druckausgaben,
- Formularunterstützung,
- Bedienerführung innerhalb des SINIX-Bedien-systems für Administrator und Benutzer,
- Unterstützung der Druckersteuersprachen HP LaserJet II*) und PostScript,
- Unterstützung von Druckern, die über LAN angeschlossen sind (Terminalserver TACLAN)
- Seitengerechtes Wiederaufsetzen von Druckausgaben nach Druckerstörung (nicht bei HP Laserjet II *),

*) Die Druckersteuersprache HP LaserJet II enthält keine Statusmeldungen. Daher ist es nicht empfehlenswert, Drucker mit dieser Steuersprache als abgesetzte Abteilungsdrucker einzusetzen.

- Virtueller Drucker (zur Entlastung der Anwendungen von der Unterstützung verschiedener Druckersteuersprachen),
- Unterstützung des 8-bit-codierten Zeichensatzes nach ISO 8859-1 bei entsprechenden Druckern und
- Programmschnittstelle (C-Interface).

Verbesserung der System-Zugangskontrolle

In SINIX V5.40 wird die System-Zugangskontrolle mittels Chipkarte vorbereitet. Mit der Anschließbarkeit eines Chipkarten-Terminals und den Erweiterungen beim System-Zugang (login, su) sind die notwendigen Voraussetzungen für den Einsatz von ASECO (Advanced Security Control) geschaffen. ASECO ist ein eigenständiges Software-Produkt zur Identifikation und Authentisierung von Systembenutzern.

Höhere Verfügbarkeit - "Spiegelplatten"

SINIX V5.40 enthält wichtige Funktionen, auf deren Basis in Verbindung mit organisatorischen Maßnahmen und bei Einsatz weiterer Produkte eine höhere Systemverfügbarkeit erreicht werden kann.

Um die Integrität und Verfügbarkeit großer Datenbestände zu gewährleisten und damit die Ausfallsicherheit von Systemen deutlich zu verbessern, können "Spiegelplatten" geführt werden. Mit SINIX V5.40 wird zunächst die Spiegelung vollständiger Platten-Partitionen unterstützt. Damit können Anwendungen, die unter Umgehung des UNIX Dateisystems direkt auf der Platte arbeiten (z.B. INFORMIX-TURBO) von dieser Funktion Gebrauch machen. Die Führung von Spiegelplatten ist für die Anwendung vollständig transparent. Für die Administration werden entsprechende Kommandos für die Konfiguration, Backup und Statusabfragen bereitgestellt.

Bei Ausfall eines Plattenlaufwerks bzw. eines Platten-Controllers wird somit der Zugriff auf einen konsistenten Datenbestand und die reibungslose Fortsetzung des Betriebs ermöglicht.

C-Entwicklungssystem (CES)

Das C-Entwicklungssystem CES ist im Lieferumfang enthalten.

CES trägt bei der Softwareentwicklung durch seinen reichhaltigen Werkzeugsatz wesentlich zur Steigerung der Qualität und Herabsetzung des Programmieraufwandes bei. Ein Großteil der CES-Werkzeuge ist auch sprachunabhängig einsetzbar. Neben dem C-Compiler und den üblichen Standardwerkzeugen wie Bibliotheksverwalter und Binder sind Funktionen enthalten, die über die Entwicklung hinaus auch die Handhabung und Pflege großer Programmpakete wirkungsvoll unterstützen (Make und Source Code Control System SCCS).

Außerdem sind enthalten:

eine Testhilfe (Debugger), ein Parser-Generator und ein Programm zur lexikalischen Analyse zur Unterstützung der Implementierung anderer Sprachen.

Für den C-Compiler im CES wird weitgehende Übereinstimmung mit dem ANSI C Draft Standard erreicht.

Eine wesentliche Verbesserung in der SW-Entwicklung wird erreicht durch:

- Dynamic Linking & Shared Libraries

Dabei werden die benötigten Bibliotheksfunktionen erst bei Programmablauf bei Bedarf dazugebunden. Ihr Code wird nicht zum Programmcode kopiert, sondern er ist nur einmal im System vorhanden und wird von allen abhängigen Anwendungen gemeinsam benutzt. Der Datenbereich wird allerdings einmal pro Prozeß angelegt.

Vorteile: kleinere Objektprogramme, weniger Paging und damit besserer Systemdurchsatz sowie geringerer Plattenspeicherbedarf; allgemein benutzte Bibliotheken können leicht ausgetauscht werden (Versionswechsel).

- Memory-Mapped Files

Um aus Programmen auf Dateien bzw. Geräte-Ein/Ausgabe zugreifen zu können, sind bisher explizite Systemaufrufe sowie das sequentielle Lesen bzw. Schreiben und ein evtl. Puffern von Daten erforderlich. Durch das Memory Mapping von Dateien (und Gerätedateien) wird der Dateiinhalt in den Prozeß-Adreßraum eingefügt und ist in gleicher Weise zugreifbar und strukturierbar wie die übrigen Programmdateien, die über Variablen adressiert werden.

Dies bewirkt eine wesentliche Vereinfachung bei der Software-Erstellung sowie mehr Effizienz beim Programmablauf.

- Bourne-, C- und Korn-Shell

sind als wichtige Funktionen für Entwickler bzw. "Experten" gemeinsam enthalten. Die Verfügbarkeit aller drei Shells bietet Freiheit bei der Auswahl des jeweils besser geeigneten bzw. bevorzugten Werkzeugs sowie Portabilität für Shell-Scripts.

Technische Daten

Hardware:

MX 300-45*/-50/-60

* Das HW-/SW-Paket MX300 beinhaltet nur das Laufzeitsystem von SINIX V5.40. In diesem sind nicht enthalten: CES, „Spiegelplatten“ und die Anschlußfähigkeit bestimmter Peripheriegeräte.



Verifizierung für BASE XPG3 und
PLUS XPG3 in Vorbereitung

Herausgegeben vom Bereich
Daten- und Informationstechnik
Postfach 83 09 51, D-8000 München 83

Siemens Aktiengesellschaft

Speicherbedarf

Hauptspeicher: 8 Mbyte;
Plattenspeicher: ca. 70 Mbyte

Nicht benötigte Funktionspakete können in gewissem Umfang vom Anwender deinstalliert werden.

Implementierungssprache:

C, Assembler

Bedienoberfläche:

Sprache: Deutsch und Englisch standardmäßig.

Installation

Grundsätzlich sind die Freigabemitteilung und das Systemverwalterhandbuch zu beachten.

Dokumentation

Benutzerhandbücher (optional)

Schulung

Kurse werden an der Schule für Daten- und Informationstechnik in München und weiteren Regionalschulen durchgeführt. Bei Bedarf können auch Kurse vor Ort vereinbart werden.

Konditionen

Dieses Softwareprodukt wird den Kunden zu unseren Bedingungen für die Nutzung von Softwareprodukten gegen einmaliges Entgelt überlassen.

Gewährleistung:

Klasse: A

Auslieferungsformat: Maschinsprache

Bestell- und Lieferhinweise

Das Softwareprodukt kann über die zuständige Siemens Data- oder Landesgesellschaft bzw. Zweigniederlassung bezogen werden.

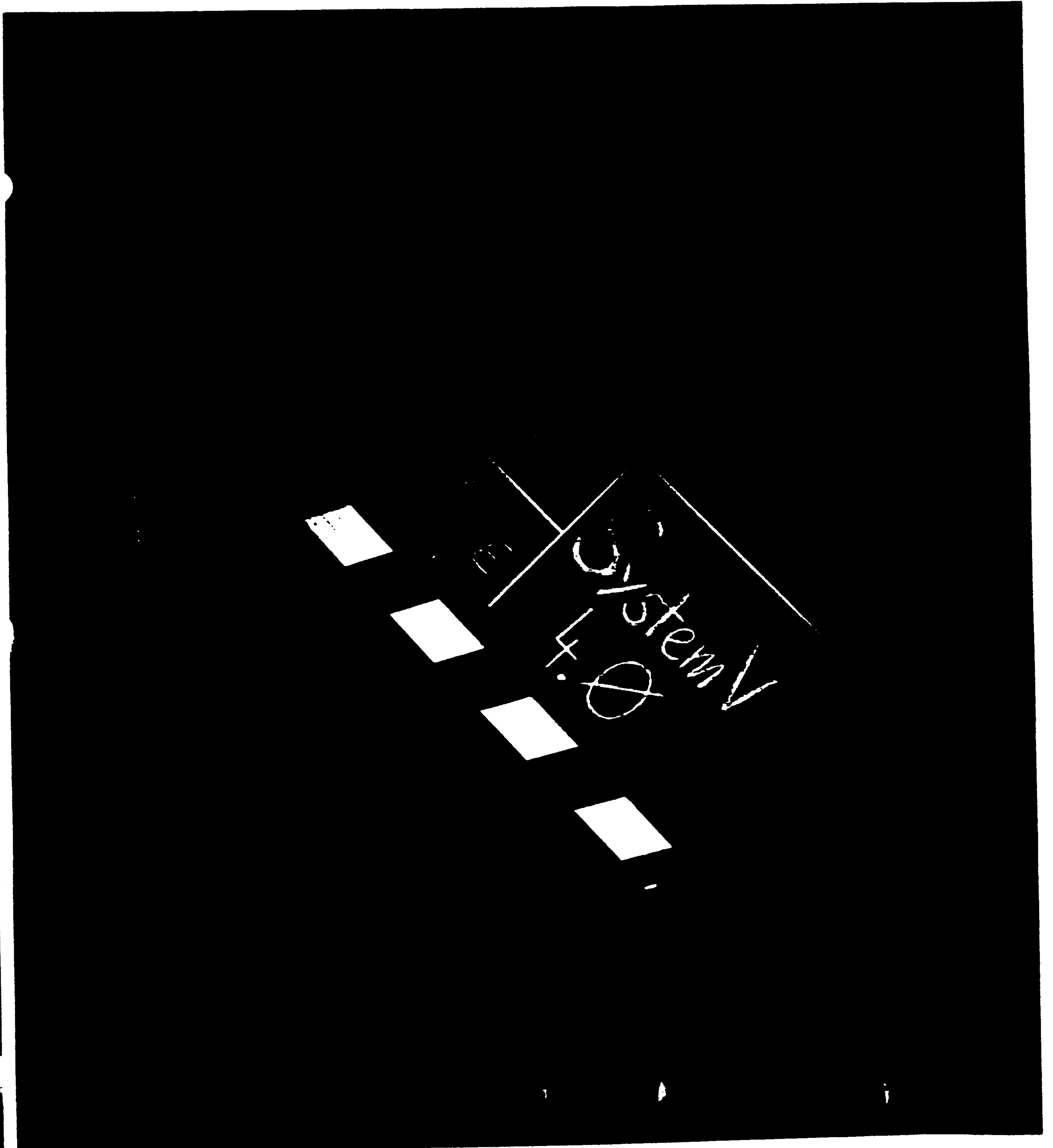
UNIX ist ein eingetragenes Warenzeichen von AT&T
X/Open ist ein eingetragenes Warenzeichen der X/Open Company Ltd

Alle Rechte vorbehalten, insbesondere für den Fall der Patenterteilung oder GM-Eintragung.
Liefermöglichkeiten und technische Änderungen vorbehalten
Copyright © Siemens AG 1990

Bestell-Nr. U5061-J-Z90-4
Printed in the Federal Republic of Germany
100PA7901.0(130)

Neue Einheit

System V Release 4.0 – erste praktische Eindrücke



1

2

3

4

Jörg Hattenhauer, Michael Klenke

Nach langem Warten ist das neue Release von SunOS 4.0 erschienen. In diesem Release wird das alte System V vereint. All diese Unix-Varianten haben ihren Ursprung in der Version 7 des von Thomson und Ritchie entworfenen Ur-Unix. Nach einigen Jahren der Evolution haben sich die Derivate jedoch immer weiter voneinander entfernt, und der gemeinsame Urahn läßt sich teilweise nur noch 'erahnen' [3]. Welcher Programmierer kennt nicht die Vielzahl der Compiler-Defines, die er in seinem 'portablen' Programmcode unterbringen muß, damit sein Programm auf möglichst vielen Maschinen lauffähig ist. Mit Release 4.0, dem ersten Produkt unter der Regie von AT&Ts eigener Unix-Abteilung (USO) soll diesem Umstand ein Ende bereitet werden.

Unix-Systeme, die in der Vergangenheit die Eigenschaften von System V, System 7, System 8, System 9, System 10, System 11, System 12, System 13, System 14, System 15, System 16, System 17, System 18, System 19, System 20, System 21, System 22, System 23, System 24, System 25, System 26, System 27, System 28, System 29, System 30, System 31, System 32, System 33, System 34, System 35, System 36, System 37, System 38, System 39, System 40, System 41, System 42, System 43, System 44, System 45, System 46, System 47, System 48, System 49, System 50, System 51, System 52, System 53, System 54, System 55, System 56, System 57, System 58, System 59, System 60, System 61, System 62, System 63, System 64, System 65, System 66, System 67, System 68, System 69, System 70, System 71, System 72, System 73, System 74, System 75, System 76, System 77, System 78, System 79, System 80, System 81, System 82, System 83, System 84, System 85, System 86, System 87, System 88, System 89, System 90, System 91, System 92, System 93, System 94, System 95, System 96, System 97, System 98, System 99, System 100, System 101, System 102, System 103, System 104, System 105, System 106, System 107, System 108, System 109, System 110, System 111, System 112, System 113, System 114, System 115, System 116, System 117, System 118, System 119, System 120, System 121, System 122, System 123, System 124, System 125, System 126, System 127, System 128, System 129, System 130, System 131, System 132, System 133, System 134, System 135, System 136, System 137, System 138, System 139, System 140, System 141, System 142, System 143, System 144, System 145, System 146, System 147, System 148, System 149, System 150, System 151, System 152, System 153, System 154, System 155, System 156, System 157, System 158, System 159, System 160, System 161, System 162, System 163, System 164, System 165, System 166, System 167, System 168, System 169, System 170, System 171, System 172, System 173, System 174, System 175, System 176, System 177, System 178, System 179, System 180, System 181, System 182, System 183, System 184, System 185, System 186, System 187, System 188, System 189, System 190, System 191, System 192, System 193, System 194, System 195, System 196, System 197, System 198, System 199, System 200, System 201, System 202, System 203, System 204, System 205, System 206, System 207, System 208, System 209, System 210, System 211, System 212, System 213, System 214, System 215, System 216, System 217, System 218, System 219, System 220, System 221, System 222, System 223, System 224, System 225, System 226, System 227, System 228, System 229, System 230, System 231, System 232, System 233, System 234, System 235, System 236, System 237, System 238, System 239, System 240, System 241, System 242, System 243, System 244, System 245, System 246, System 247, System 248, System 249, System 250, System 251, System 252, System 253, System 254, System 255, System 256, System 257, System 258, System 259, System 260, System 261, System 262, System 263, System 264, System 265, System 266, System 267, System 268, System 269, System 270, System 271, System 272, System 273, System 274, System 275, System 276, System 277, System 278, System 279, System 280, System 281, System 282, System 283, System 284, System 285, System 286, System 287, System 288, System 289, System 290, System 291, System 292, System 293, System 294, System 295, System 296, System 297, System 298, System 299, System 300, System 301, System 302, System 303, System 304, System 305, System 306, System 307, System 308, System 309, System 310, System 311, System 312, System 313, System 314, System 315, System 316, System 317, System 318, System 319, System 320, System 321, System 322, System 323, System 324, System 325, System 326, System 327, System 328, System 329, System 330, System 331, System 332, System 333, System 334, System 335, System 336, System 337, System 338, System 339, System 340, System 341, System 342, System 343, System 344, System 345, System 346, System 347, System 348, System 349, System 350, System 351, System 352, System 353, System 354, System 355, System 356, System 357, System 358, System 359, System 360, System 361, System 362, System 363, System 364, System 365, System 366, System 367, System 368, System 369, System 370, System 371, System 372, System 373, System 374, System 375, System 376, System 377, System 378, System 379, System 380, System 381, System 382, System 383, System 384, System 385, System 386, System 387, System 388, System 389, System 390, System 391, System 392, System 393, System 394, System 395, System 396, System 397, System 398, System 399, System 400, System 401, System 402, System 403, System 404, System 405, System 406, System 407, System 408, System 409, System 410, System 411, System 412, System 413, System 414, System 415, System 416, System 417, System 418, System 419, System 420, System 421, System 422, System 423, System 424, System 425, System 426, System 427, System 428, System 429, System 430, System 431, System 432, System 433, System 434, System 435, System 436, System 437, System 438, System 439, System 440, System 441, System 442, System 443, System 444, System 445, System 446, System 447, System 448, System 449, System 450, System 451, System 452, System 453, System 454, System 455, System 456, System 457, System 458, System 459, System 460, System 461, System 462, System 463, System 464, System 465, System 466, System 467, System 468, System 469, System 470, System 471, System 472, System 473, System 474, System 475, System 476, System 477, System 478, System 479, System 480, System 481, System 482, System 483, System 484, System 485, System 486, System 487, System 488, System 489, System 490, System 491, System 492, System 493, System 494, System 495, System 496, System 497, System 498, System 499, System 500, System 501, System 502, System 503, System 504, System 505, System 506, System 507, System 508, System 509, System 510, System 511, System 512, System 513, System 514, System 515, System 516, System 517, System 518, System 519, System 520, System 521, System 522, System 523, System 524, System 525, System 526, System 527, System 528, System 529, System 530, System 531, System 532, System 533, System 534, System 535, System 536, System 537, System 538, System 539, System 540, System 541, System 542, System 543, System 544, System 545, System 546, System 547, System 548, System 549, System 550, System 551, System 552, System 553, System 554, System 555, System 556, System 557, System 558, System 559, System 560, System 561, System 562, System 563, System 564, System 565, System 566, System 567, System 568, System 569, System 570, System 571, System 572, System 573, System 574, System 575, System 576, System 577, System 578, System 579, System 580, System 581, System 582, System 583, System 584, System 585, System 586, System 587, System 588, System 589, System 590, System 591, System 592, System 593, System 594, System 595, System 596, System 597, System 598, System 599, System 600, System 601, System 602, System 603, System 604, System 605, System 606, System 607, System 608, System 609, System 610, System 611, System 612, System 613, System 614, System 615, System 616, System 617, System 618, System 619, System 620, System 621, System 622, System 623, System 624, System 625, System 626, System 627, System 628, System 629, System 630, System 631, System 632, System 633, System 634, System 635, System 636, System 637, System 638, System 639, System 640, System 641, System 642, System 643, System 644, System 645, System 646, System 647, System 648, System 649, System 650, System 651, System 652, System 653, System 654, System 655, System 656, System 657, System 658, System 659, System 660, System 661, System 662, System 663, System 664, System 665, System 666, System 667, System 668, System 669, System 670, System 671, System 672, System 673, System 674, System 675, System 676, System 677, System 678, System 679, System 680, System 681, System 682, System 683, System 684, System 685, System 686, System 687, System 688, System 689, System 690, System 691, System 692, System 693, System 694, System 695, System 696, System 697, System 698, System 699, System 700, System 701, System 702, System 703, System 704, System 705, System 706, System 707, System 708, System 709, System 710, System 711, System 712, System 713, System 714, System 715, System 716, System 717, System 718, System 719, System 720, System 721, System 722, System 723, System 724, System 725, System 726, System 727, System 728, System 729, System 730, System 731, System 732, System 733, System 734, System 735, System 736, System 737, System 738, System 739, System 740, System 741, System 742, System 743, System 744, System 745, System 746, System 747, System 748, System 749, System 750, System 751, System 752, System 753, System 754, System 755, System 756, System 757, System 758, System 759, System 760, System 761, System 762, System 763, System 764, System 765, System 766, System 767, System 768, System 769, System 770, System 771, System 772, System 773, System 774, System 775, System 776, System 777, System 778, System 779, System 780, System 781, System 782, System 783, System 784, System 785, System 786, System 787, System 788, System 789, System 790, System 791, System 792, System 793, System 794, System 795, System 796, System 797, System 798, System 799, System 800, System 801, System 802, System 803, System 804, System 805, System 806, System 807, System 808, System 809, System 810, System 811, System 812, System 813, System 814, System 815, System 816, System 817, System 818, System 819, System 820, System 821, System 822, System 823, System 824, System 825, System 826, System 827, System 828, System 829, System 830, System 831, System 832, System 833, System 834, System 835, System 836, System 837, System 838, System 839, System 840, System 841, System 842, System 843, System 844, System 845, System 846, System 847, System 848, System 849, System 850, System 851, System 852, System 853, System 854, System 855, System 856, System 857, System 858, System 859, System 860, System 861, System 862, System 863, System 864, System 865, System 866, System 867, System 868, System 869, System 870, System 871, System 872, System 873, System 874, System 875, System 876, System 877, System 878, System 879, System 880, System 881, System 882, System 883, System 884, System 885, System 886, System 887, System 888, System 889, System 890, System 891, System 892, System 893, System 894, System 895, System 896, System 897, System 898, System 899, System 900, System 901, System 902, System 903, System 904, System 905, System 906, System 907, System 908, System 909, System 910, System 911, System 912, System 913, System 914, System 915, System 916, System 917, System 918, System 919, System 920, System 921, System 922, System 923, System 924, System 925, System 926, System 927, System 928, System 929, System 930, System 931, System 932, System 933, System 934, System 935, System 936, System 937, System 938, System 939, System 940, System 941, System 942, System 943, System 944, System 945, System 946, System 947, System 948, System 949, System 950, System 951, System 952, System 953, System 954, System 955, System 956, System 957, System 958, System 959, System 960, System 961, System 962, System 963, System 964, System 965, System 966, System 967, System 968, System 969, System 970, System 971, System 972, System 973, System 974, System 975, System 976, System 977, System 978, System 979, System 980, System 981, System 982, System 983, System 984, System 985, System 986, System 987, System 988, System 989, System 990, System 991, System 992, System 993, System 994, System 995, System 996, System 997, System 998, System 999, System 1000.

Unix-Varianten vereinen soll. Welche Vorteile das bringt, ob diese Lösung den 'Glaubenskrieg' zwischen BSDlern und SYSVlern beenden kann oder ob neue Probleme aufgeworfen werden – erste Eindrücke nach einem Blick auf eine real existierende Implementierung auf ICLs DRS 6000.

In AT&Ts Unix-System V Release 4.0 sind SunOS, Xenix, BSD und das 'alte' System V vereint. All diese Unix-Varianten haben ihren Ursprung in der Version 7 des von Thomson und Ritchie entworfenen Ur-Unix. Nach einigen Jahren der Evolution haben sich die Derivate jedoch immer weiter voneinander entfernt, und der gemeinsame Urahn läßt sich teilweise nur noch 'erahnen' [3]. Welcher Programmierer kennt nicht die Vielzahl der Compiler-Defines, die er in seinem 'portablen' Programmcode unterbringen muß, damit sein Programm auf möglichst vielen Maschinen lauffähig ist. Mit Release 4.0, dem ersten Produkt unter der Regie von AT&Ts eigener Unix-Abteilung (USO) soll diesem Umstand ein Ende bereitet werden.

Aus vier mach eins

Eine andere Zielsetzung bei der Entwicklung war der Bedarf nach Standardisierung. In der Release 4.0 hat man versucht, sich nach mittlerweile definierten Standards zu richten. So sind viele Teile konform zur POSIX-Norm P1003, zum Beispiel File-Manipulationen oder Job Control Shell. In den Bereichen, in denen noch keine Standards feststehen, wurden De-facto-Standards aus bestehenden Unix-Varianten zugrunde gelegt. Und da man ja selbst auch ein wenig Standard ist, wurde die allseits bekannte SVID (System V Interface De-

inition) ebenfalls den neuen Gegebenheiten angepaßt.

Files mit System

Dieses Bestreben nach Vereinigung und Standardisierung erforderte ein weitgehendes Redesign einiger Teile des Unix-Systems. Als Beispiel sei hier der Übergang zu einem virtuellen Filesystem (VFS) genannt – ein notwendiger Schritt, da eine Vielzahl von verschiedenen Filesystem-Typen existieren, die es alle zu unterstützen gilt. Allein unter den älteren Versionen von System V gibt es schon drei verschiedene Typen (0.5-, 1- und 2-KByte Blocksize), weitere kommen in Release 4.0 dazu (siehe Kasten 'Die System-V4.0-Filesysteme'). Um einen vom Filesystem-Typ unabhängigen Zugang anzubieten, schaltete man VFS dazwischen. Hier findet der Benutzer eine einheitliche Schnittstelle zu den darunterliegenden Filesystemen. Spezielle Treiber im Kernel sorgen dafür, daß die Anfragen in typabhängige Funktionen umgesetzt werden. Es ist auch möglich, diese Treiber um eigene Typen zu erweitern und so neue Filesysteme hinzuzufügen (ähnlich dem Konzept von Device-Treibern).

Systemübergreifend einheitlich ist das Boot Files System (BFS). Es enthält alle zum Booten eines Rechners benötigten Programme inklusive des Betriebssystemkerns. Das Bootstrap-Programm hat sich so nicht mit den vielen möglichen Filesystem-Typen herumzu-

schlagen, sondern muß nur den Typ BFS kennen und ist daher weniger komplex. Das hat auch den Vorteil, daß der Rechner von jedem beliebigen Device, das von der Firmware angesprochen werden kann und ein Filesystem im BFS-Format enthält, gebootet werden kann. Dazu legt das Installations-Skript eine eigene Partition *stand* an, auf der ein BFS mit den zum Booten benötigten Programmen erzeugt wird.

Damit das Betriebssystem auch 'weiß', auf welcher Partition sich welcher Typ von Filesystem befindet, muß in der *fstab*, die nun *vfstab* heißt, angegeben werden, welche Partition wie zu behandeln ist. Eine *vfstab* ähnelt dann etwa dem Beispiel im Kasten '/etc/vfstab'.

Zu den bekannten Feldern sind noch einige neue Felder hinzugekommen. Das vierte Feld gibt an, um welchen Filesystem-Typ es sich handelt. Dies ist nötig, damit zum Mounten der passende Treiber im Kernel aktiviert werden kann. Feld 6 teilt dem *mount*-Kommando mit, ob dieses Filesystem bei der Option '-a' gemountet werden soll. Auf diese Weise kann man eine ausführliche *fstab* erstellen, mit der bei Bedarf der Administrator einzelne Filesysteme einhängen kann.

Eine weitere Neuerung ist das *proc*-Filesystem, auf das wir später noch näher eingehen werden.

Bäumchen, wechsele dich

Auch der Dateibaum wurde unter Release 4 neu gestaltet, um eine Struktur zu erhalten, mit der man recht einfach Partitions netzwerkweit anbieten kann. Die neue Struktur lehnt sich stark an die unter SunOS 4.X (siehe [4]) verwendete an. So gibt es eine Aufteilung in *root*-, *usr*-, *var*- und *home*-Pfade. Das Root-Directory enthält nach wie vor die für das

Booten notwendigen Files. Das *usr*-Directory hingegen beinhaltet nur noch die Files, die sich während des Betriebs nicht mehr ändern. Unter *usr/share* befinden sich alle architekturunabhängigen Dateien, die über ein Netzwerk auch anderen Rechnern zugänglich gemacht werden sollen, etwa die Online-Handbücher. Alle Files, die sich während der Laufzeit verändern, wie Log-Files oder ähnliches, residieren im *var*-Directory. So findet man unter *var* Subdirectories wie *mail*, *preserve*, *spool* oder *tmp*. Um Umsteigern auf diese neue Struktur das Leben nicht unnötig schwer zu machen, sind diese Files zusätzlich durch symbolische Links unter ihrem alten Pfad zu erreichen. Unter *home* schließlich stehen die Heimatverzeichnisse der einzelnen Benutzer.

Kommandobrücke

Der Standardisierungsprozeß vereinigte viele bislang unter System V unbekannte Kommandos in SYSVR4: vieles aus BSD- oder Xenix-Umgebungen, vom BSD wurden die meisten Netzwerkkommandos übernommen. Natürlich bringt es Probleme mit sich, wenn man verschiedene Kommandomenge vereinigen will – verschiedene Befehle haben den gleichen Namen, Konflikte mit bestehenden Standards et cetera. In Release 4 hat man sich entschieden. Befehlsnamen sowie deren Optionen nur zu ändern, falls sich eine Notwendigkeit dafür ergibt. Kommandos mit derselben Funktionalität aus verschiedenen Unix-Linien sind zu einem Kommando zusammengefaßt worden. Allerdings verzichtete man auf einige etwas weniger bekannte Kommandos oder solche, die sich zu stark von ihren System-V-Kollegen unterscheiden. Die wurden in sogenannte Kompatibilitätsdirectories verlagert, etwa *usr/uch*.

/etc/vfstab – fstab in SV4.0

#specdev	fsckdev	mountpt	fstype	ckpass	auto	ro
boot	-	boot	boot	-	-	-
dev fs	-	dev fs	fs	-	-	-
dev root	dev root	root	root	-	-	-
dev:ask:01d0s0	dev:ask:01d0s0	stand	ufs	-	-	-
dev:ask:01d0s9	dev:ask:01d0s9	usr	ufs	-	-	-
dev:ask:01d0s9	dev:ask:01d0s9	var	ufs	-	-	-
dev:ask:01d0s10	dev:ask:01d0s10	home	ufs	-	-	-

Dem Benutzer präsentiert sich das neue Unix im neuen Gewand: Die Job Control Shell enthält die gleichen Job-Kontroll-Mechanismen wie die v.a. jedoch alles POSIX-1003.1-konform.

Standard-GUI

Ein weiteres Beispiel für die Integration von De-facto-Standards ist die Auswahl der grafischen Oberflächen. Release 4 bietet ein device-unabhängiges und portables Window-System an, das 'Graphical User Interface' (GUI). Das GUI vereinigt XWIN, X11/NeWS und Open Look. (XWIN ist im Prinzip das 'normale' X-Window-System, X11/NeWS eine Umgebung, die sowohl das X11 als auch Suns NeWS-Protokoll unterstützt, Open Look ein Standard für das 'Look&Feel' von grafischen Fensteroberflächen.) Die Integration geht so weit, daß man mit einem einzigen Window-Manager auskommt.

Internationalization

Unix hat sich mittlerweile nicht nur in akademischen Umgebungen etabliert, sondern auch in vielen kommerziellen Bereichen durchgesetzt. Gerade im Anwendungssektor wird der Benutzerfreundlichkeit eine große Bedeutung beigemessen: Applikationen sollten auch von unbedarften Benutzern ohne Probleme angewandt werden können. Fremdsprachige Erklärungen stellen sich hier oft als unnütze Hürde dar. Obwohl man dieses erkannt hat und Handbücher schon gelegentlich in der jeweiligen Landessprache mitliefert, werden nur wenige landessprachliche Programmversionen angeboten. Hier spielt sich fast alles noch auf Englisch ab.

Mittlerweile existieren im PC-/MSDOS-Bereich bereits einige deutsche Versionen bekannter Programme, die oft den englischsprachigen Originalen vorgezogen werden. Das muß aber nicht verschiedene, auf die jeweiligen Länder angepaßte Versionen zur Folge haben, es reicht vollkommen, nur die Texte entsprechend auszutauschen. Dieser Weg wird in System V Release 4 eingeschlagen. Um ein Unix zu entwickeln, das unabhängig von

Benutzersprache und Einsatzland ist, sind Funktionen implementiert, die dies schon seitens des Betriebssystems unterstützen sollen.

Ein Beispiel für solch eine 'table-look-up'-Funktion ist *gettext*. Man setzt sich mit Hilfe der Funktion *setlocale* die gewünschte lokale Umgebung, und *gettext* liefert dann den entsprechenden Text in der lokalen Sprache zurück. Eine weitere landesspezifische Anforderung ist das Datumsformat, das sich im Deutschen stark von dem in englischsprachigen Umgebungen verwendeten Format unterscheidet. Probleme für den lokalen Einsatz können sich auch aus dem verwendeten Zeichensatz ergeben. Weit verbreitet sind der 7-Bit-ASCII-Code oder die 8-Bit-European Codes (mit Sonderzeichen). Dieses reicht aber nicht für alle Anwender, das japanische Kanji etwa benötigt bis zu 3 Bytes pro Zeichen. Dieses erfordert Unterschiede zwischen den Datentypen Byte und Char, was bis jetzt durchaus nicht so eng gegeneinander abgegrenzt war. Viele dieser Ideen stammen aus den ANSI- und POSIX-Standarddefinitionen. (ANSI X3.159-1989, IEEE Std. 1003.1).

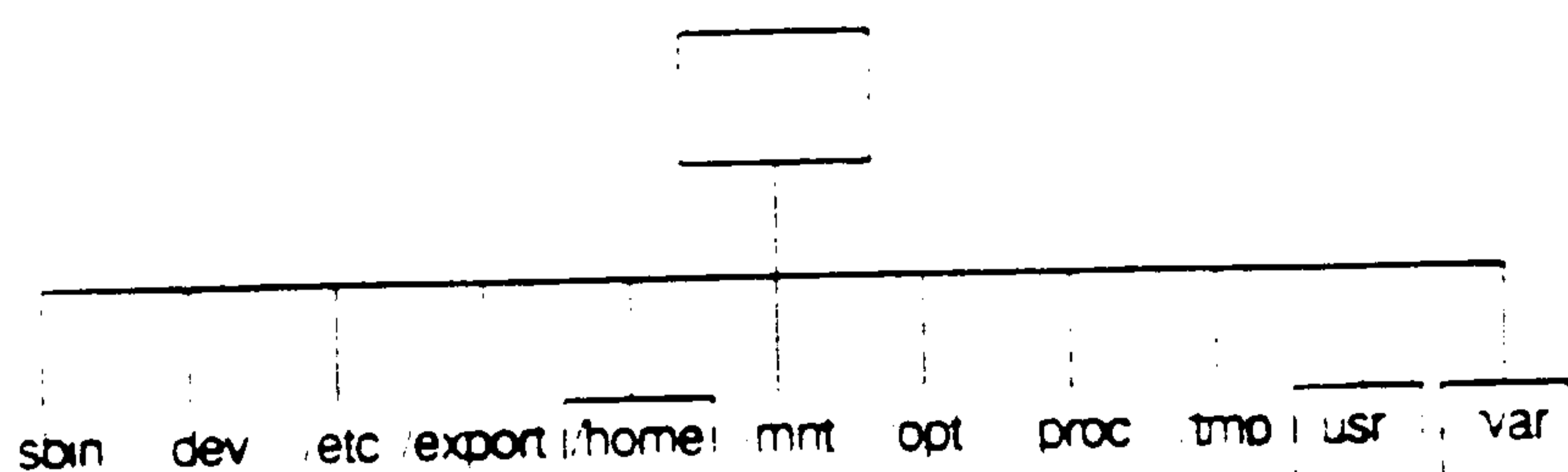
Backups erleichtert

Ein leidiges Problem war bei allen Unix-Maschinen bisher das Thema Backup. Die meisten Hersteller ließen ihre Kunden in diesem wichtigen Punkt im Stich. Archaische Utilities wie *dump* und *restore* mußten herhalten, wenn nicht die Zeit vorhanden war, ein eigenes System zu entwickeln. Die Zahl solcher selbstgestrickten Utilities ist sicherlich Legion. Dieses Manko soll mit dem neuen Unix behoben sein. Um dem Benutzer ein mächtiges und auf allen Maschinen einheitliches System zur Verfügung zu stellen, bietet Release 4.0 ein eingebautes Backup-System an.

Wichtigster Bestandteil dieses Systems ist das sogenannte Backup-Register. Dahinter verbirgt sich eine Datei, in der genau vermerkt ist, welche Dateien oder Filesysteme zu einer bestimmten Zeit zu sichern sind. Innerhalb einer vom Benutzer festzulegenden Backup-

Die System V File Directory

Das Root-Filesystem



<i>sbin</i>	Zentrale Programme für Administration und Operating
<i>dev</i>	Character- und Block-Special-Files
<i>etc</i>	Maschinen-spezifische Konfigurations-Files und Daten zur Systemadministration (keine ausführbaren Dateien)
<i>export</i>	Default-Wurzel für exportierte Dateibäume
<i>home</i>	Standard-Filesystem für Benutzerverzeichnisse
<i>mnt</i>	Default für temporäres Mounten
<i>opt</i>	Unterverzeichnisse für Anwenderprogramme
<i>proc</i>	Prozeß-Filesystem
<i>tmp</i>	Temporäre Dateien des Systems
<i>usr</i>	Standard-Filesystem für dauerhafte, 'shareable' Dateien
<i>var</i>	Standard-Filesystem für sich verändernde Dateien

Periode können verschiedene 'Methoden' zur Sicherung der Daten aufgerufen werden. Setzt man diese Backup-Periode beispielsweise auf vier Wochen fest, so kann man die einzelnen Tage mit einem Paar 'X:Y' festlegen. 'X' steht dabei für die Woche und 'Y' für den Tag in dieser Woche. Die Methoden geben an, welche Dateien aus dem angegebenen Filesystem zu sichern sind. Die Methode *incfile* etwa steht für eine inkrementelle Sicherung. Andere Methoden sind *fimage* oder *ffile*. Jede dieser Sicherungen wird mit einer Backup-ID versehen und in einer Datenbank festgehalten. Ausnahmelisten schließen nicht zu sichernde Files aus. Ein Restaurieren der Daten erfolgt dann anhand der Backup-ID.

Alle diese Kommandos können sowohl interaktiv als auch im Hintergrund ablaufen. Wird bei einem Hintergrund-Backup einmal die Anwesenheit eines Operators nötig, erhält dieser eine Nachricht per Mail.

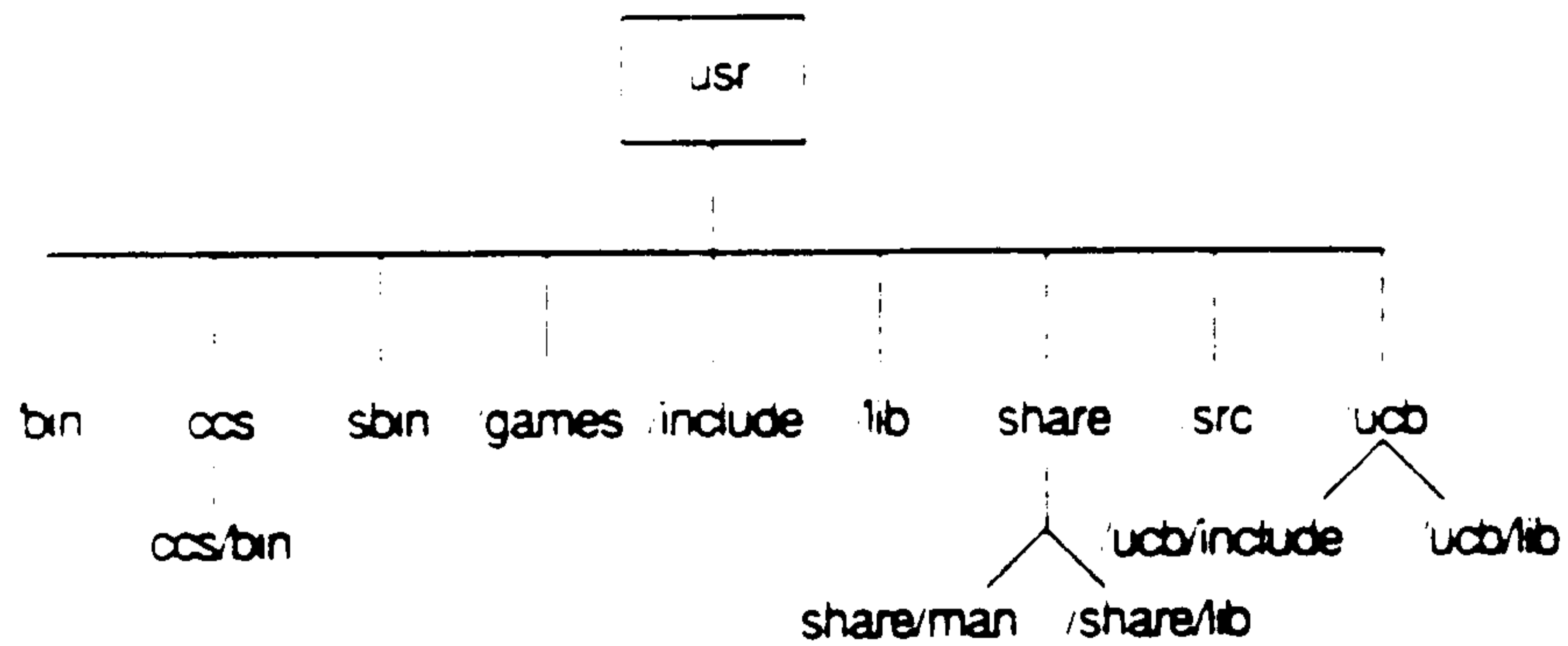
Ein sinnvolles Feature ist die Möglichkeit, Backups von einem Medium auf ein anderes zu verlagern (migration). Sind

auf einem Tape mehrere Backups und besitzt davon nur noch eins Relevanz für die aktuelle Backup-Periode, läßt sich dieses auf ein anderes Tape spielen. Das alte Tape wird eingelesen und der relevante Backup auf ein anderes Tape übertragen. Damit *restore* nicht durcheinander kommt, werden die Protokolldateien aktualisiert, und das alte Tape steht wieder ganz zur Verfügung.

ABI oder Mittlere Reife

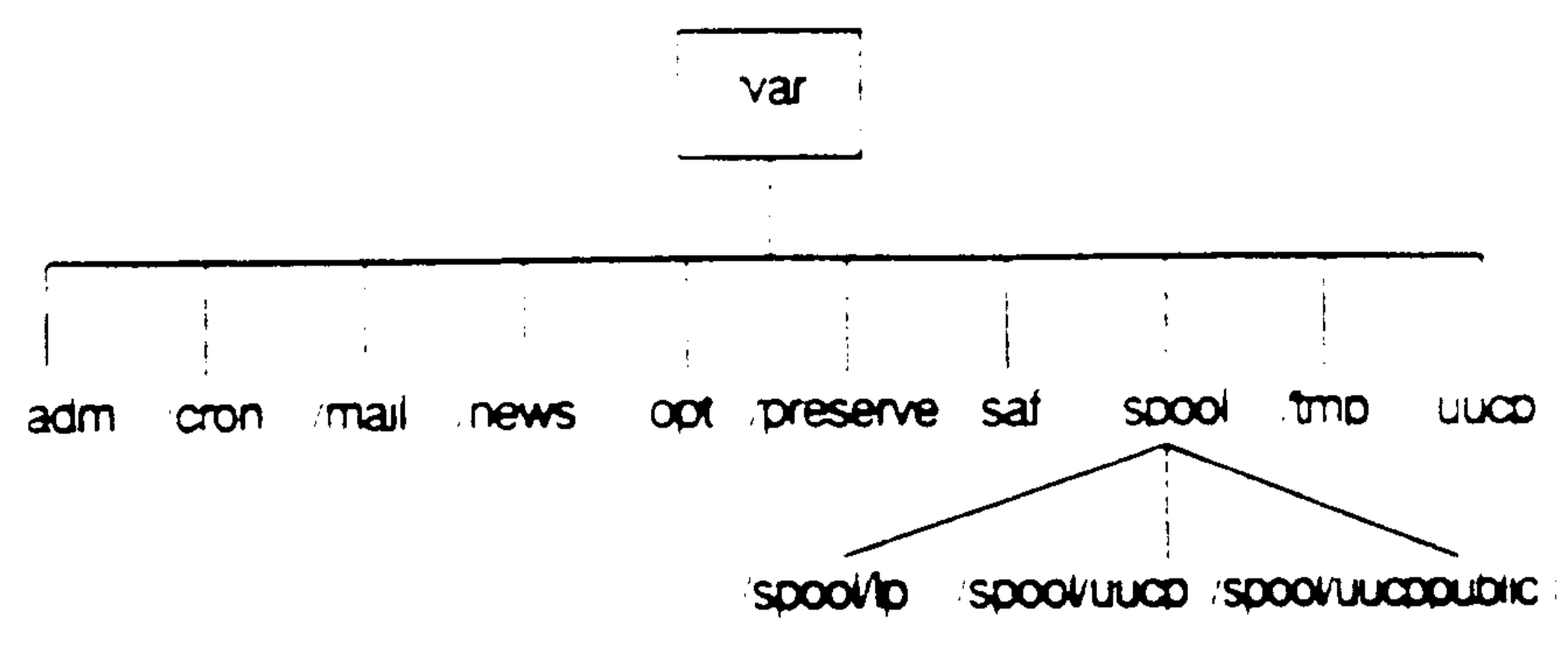
Eine weitere Standardisierung von Unix vorantreiben soll das Application Binary Interface, kurz ABI. Das ABI ist eine festgelegte Auswahl von System-Calls, Library-Routinen und Programmen mit dem Ziel, eine unter Unix erstellte Applikation portabler zu gestalten. Jeder Unix-Rechner, der diesem Standard entspricht, muß diese festgelegte Schnittstelle zur Verfügung stellen. Programme, die sich an das ABI halten, finden so auf jedem Rechner mit ABI einen fest definierten Funktionsumfang vor und müssen auf anderen Rechnern höchstens erneut übersetzt werden. Bei Rechnern

Das /usr-Filesystem



<code>/usr/bin</code>	Der überwiegende Teil der System-Utilities
<code>/usr/ocs</code>	C-Directory
<code>/usr/ocs/bin</code>	System-Utilities für C
<code>/usr/sbin</code>	Ausführbare Dateien für die System-Administration
<code>/usr/games</code>	Spiele
<code>/usr/include</code>	Header-Files
<code>/usr/lib</code>	Libraries und architekturabhängige Daten
<code>/usr/share</code>	Architekturunabhängige, 'shareable' Dateien
<code>/usr/share/man</code>	Online-Manuals
<code>/usr/share/lib</code>	Architekturunabhängige Daten (z. B. Textdateien)
<code>/usr/src</code>	Source-Code für Utilities und Libraries (nur bei Source-Code-Lizenz)
<code>/usr/ucb</code>	BSD-kompatible Binaries
<code>/usr/ucb/include</code>	BSD-kompatible Header-Files
<code>/usr/ucb/lib</code>	BSD-kompatible Libraries

Das /var-Filesystem



<code>/var/adm</code>	Log- und Accounting-Files des Systems
<code>/var/cron</code>	Log-Files für <i>cron</i>
<code>/var/mail</code>	Mail-Dateien der Benutzer
<code>/var/news</code>	Nachrichten von allgemeinem Interesse
<code>/var/options</code>	Enthält eine Datei zur Identifizierung jedes installierten Utility
<code>/var/preserve</code>	Backups für <i>vi</i> und <i>ex</i>
<code>/var/saf</code>	Log- und Accounting-Files der Service Access Facility
<code>/var/spool</code>	Temporäre Spool-Dateien
<code>/var/spool/lp</code>	Drucker-Spool-Dateien
<code>/var/spool/uucp</code>	Warteschlange der <i>uucp</i> -Jobs
<code>/var/spool/uucppublic</code>	Von <i>uucp</i> abgelegte Dateien
<code>/var/tmp</code>	Temporäre Files
<code>/var/uucp</code>	Log- und Accounting-Dateien für <i>uucp</i>

mit gleichem Prozessor sind sogar Binaries direkt austauschbar. Dies kommt den Programmierern und Händlern entgegen, die nicht unbedingt zu jedem Programm die gesamten Sourcen vertreiben wollen. So bricht auch für Unix das Zeitalter der 'Shrink-wrap'-Software an, wie sie in der PC-Welt seit langem bekannt ist – ohne Umwege aus dem Ladenregal in den heimischen Rechner. Viele Programme werden durch die so vereinfachte Vermarktung überhaupt erst für ein breiteres Publikum erschwinglich.

Diese beabsichtigte Vereinfachung der Portierung im Bereich kommerzieller Software-Entwicklung könnte sich dagegen im Public-Domain-Bereich eher nachteilig auswirken. Unix verdankt seine wachsende Verbreitung auch den Megabytes an Sourcen, die für jedermann frei zugänglich sind. Mit ABI besteht die Gefahr, daß in Zukunft mehr und mehr Entwickler dazu übergehen, nur noch Binaries ihrer Programme zu veröffentlichen. Dann heißt es Abschied nehmen von eigenen Weiterentwicklungen und Anpassungen an die speziellen Be-

dürfnisse. Was tatsächlich dabei herauskommt, bleibt abzuwarten.

Ein aktuelleres Problem ist die Unterstützung, die der ABI-Programmierer vom System erhält. Da er ja nur einen begrenzten Funktionsumfang seines Entwicklungssystems benutzen darf, wäre es wünschenswert, wenn es Hilfsmittel gäbe, die ihn bei seiner Programmierarbeit unterstützen. Eine Compiler-beziehungswise Linker-Option wie '-ABI' könnte darüber wachen, ob in dem Programm auch wirklich nur der ABI-Funktionsumfang ausgeschöpft wird oder ob sich eine Routine in das Programm eingeschlichen hat, die nur auf dem Entwicklungsrechner vorhanden ist. Leider sind solche Hilfen bislang nicht im Lieferumfang enthalten. Der Programmierer muß selbst darauf achten, nur ABI-konforme Funktionen zu benutzen.

Staffel-ANSI

Die Strategie der Standardisierung wurde auch bei der Wahl der Programmiersprache C konsequent beibehalten: Migration zu ANSI C. Da aber

(noch?) nicht alle C-Sourcen ANSI-konform sind, wurde der C-Compiler *cc* mit einigen neuen Optionen erweitert. Diese geben den Grad der Konformität an, das heißt, inwieweit sich der Compiler an die vorgeschriebenen Richtlinien halten soll. Vorgesehen sind drei Stufen: *transition*, *ansi* und *conformance*. Daneben wurden auch noch einige Erweiterungen hinsichtlich der Internationalisierung eingebaut. Eine genauere Darstellung der Standards findet sich in ANSI X3.159-1989.

Auch die Libraries haben sich geändert. Per Default werden jetzt dynamische Shared-Libraries unterstützt. Bei Shared-Libraries blendet das Betriebssystem die benötigten Funktionen zur Laufzeit in den virtuellen Speicher des Prozesses ein; dieses hat Auswirkungen auf die Größe eines Binaries. Weil nicht jeder Prozeß seine eigene Library besitzt, sondern sie sich mit anderen Prozessen teilt, braucht sich nur eine Kopie im Speicher zu befinden. Das spart Hauptspeicher und Plattenplatz. Der Vorteil der dynamischen Shared-Libs gegenüber den statischen liegt in der Erweiterbar-

keit der Bibliotheken, ohne gleich die Applikationen neu linkeln zu müssen. Da das alte Objektfile-Format COFF (Common Object File Format) nicht mehr angemessen war, ersetzte AT&T dieses durch ein neues, genannt ELF (Executable and Linking Format) – um den Preis der Inkompatibilität zu bisherigen System-V-Versionen.

I/O flexibler

Unix benötigt für jedes Peripheriegerät, das dem System hinzugefügt wird, einen Treiber, der im Kernel installiert wird. In vielen Fällen kommt es natürlich vor, daß Teile der Treibersoftware mehrfach vorhanden sind. Allerdings erlaubte Unix bisher nicht, bestimmte Stellen des Programmcodes für mehrere Interface-Routinen zur Verfügung zu stellen. Aus diesem Grunde wurde in Release 3 das STREAMS-Konzept eingeführt, das eine größere Modularität und Flexibilität erlaubt. STREAMS definieren ein Standard-Interface für Character-Input/Output innerhalb des Kernels, sie ermöglichen eine geräteunabhängige Formulie-

Die System-V4.0-Filesysteme

- ufs - Boot-File-System, enthält alle zum Booten benötigten Programme
- specs - enthält Devices und Special Files
- procs - Pipe Files
- fs - die drei bisherigen System-V-Filesystem-Typen (0.5K, 1K, 2K Blocksize)
- ufs - BSD-Fast-Filesystem mit 8 KByte Blocksize
- ufs - Remote File Sharing, System-V-Netzwerk-Filesystem
- ufs - Network File System, Suns Netzwerk-Filesystem
- proc - Filesystem, das Zugriff auf die laufenden Prozesse und deren Speicherbereich erlaubt

nung von High-Level-Applikationen. Release 4 erweitert dieses Konzept mit Einführung des Device Driver Interface (DDI) und des Driver Kernel Interface (DKI), die die Schnittstellen zum Kernel und zum Interface standardisieren. Wie in vielen Teilen der neuen Release hat man auch hier versucht, mit Standards die Flexibilität und Erweiterbarkeit zu erhöhen und zugleich zu erleichtern. So wurden das komplette *mv*-Subsystem und Pipes mit Hilfe von STREAMS implementiert.

3G-Memory-Management

In Release 4 wurde auf die traditionelle Swapping-Architektur zugunsten einer 'third-generation memory management architecture' à la SunOS verzichtet, um den Hauptspeicher effizienter zu gebrauchen. Als

'Nebenprodukte' fallen dann Features wie Memory-Mapped-Files, Shared Memory und Swapping in normale Files ab. Die *mmap*-Routinen erlauben es dem Applikationsprogrammierer, gewünschte Files in den Adreßraum seines Prozesses zu mappen, wo er wie bei einem Array darauf zugreifen kann. Neben den bisherigen Shared-Memory-Primitiven (*shm*), die auch weiterhin unterstützt werden, bietet Release 4 auch die Möglichkeit, Mapped-Files als Shared-Memory aufzufassen; so können mehrere Prozesse dieselbe Kopie eines Files bearbeiten. Während in bisherigen Unix-Varianten eine gesonderte Swap-Partition eingerichtet werden mußte, erlaubt man jetzt das Swappen in normale Files (ähnlich wie es Sun Microsystems bei dem NFS-Swappen der Clients in *export swap* durchführt). Durch

die strikte Trennung zwischen hardware-abhängigen und hardware-unabhängigen Memory-Management-Funktionen wird eine mögliche Portierung einfacher: auf jeder Architektur existiert dann dasselbe MMU-Interface.

UUCP ist passé

In früheren (AT&T-)Unix-Versionen war es mit der Netzwerkfähigkeit nicht sehr weit her. Die Basic Networking Utilities (BNU) bestanden hauptsächlich aus dem *uucp*-Paket. Damit konnten Point-To-Point-Verbindungen über normale Telefonleitungen hergestellt werden, etwa zum Transport von Electronic Mail. Dieses wird zwar in bestimmten Umgebungen immer noch angewandt, ist aber längst nicht mehr State of the Art.

Das ist heute die ARPAnet-Protokollfamilie TCP/IP, die durch die Integration in das Berkeley-Unix stark verbreitet wurde und heute eine der meistbenutzten Protokollfamilien ist. Die Entwickler in Berkeley entwarfen dazu das Konzept der Sockets, das Netzwerk-Applikationen ein netzunabhängiges Interface zur Verfügung stellt. Aus dem gleichen Grund führte AT&T dann in Release 3 die STREAMS und das Transport Level Interface (TLI) ein. Beides sind Hilfsmittel zur protokollunabhängigen modularen Programmierung.

Eine weitere Abstraktion stiftete Sun Microsystems in ihrem SunOS vor, die Remote Procedure Calls (RPC). Damit wird es einem lokal ablaufenden Programm möglich, Prozeduren auf einem Remote-System aufzurufen. Damit dieses auch zwischen Maschinen unterschiedlicher Architektur (etwa mit der Konsequenz verschiedener Byte-Order wie zwischen Vaxen und Suns) funktioniert, hat man auch gleich eine netzweite, architekturunabhängige Datenrepräsentation (XDR, eXternal Data Representation) geschaffen, so daß kommunizierende Prozesse sich nicht mehr um solche (hardware-abhängigen) Probleme kümmern müssen.

Sun implementierte darauf aufbauend das NFS (Network File System), ein verteiltes Filesystem, das mittlerweile nicht nur Dateien netzweit verfügbar macht, sondern auch das Betreiben von diskless Workstations, also Rechnern ohne eigene Platten, ermöglicht. Hierbei wird sogar über NFS geswappt. Unix System V Release 3 führte ebenfalls ein verteiltes Filesystem ein, das Remote File Sharing (RFS). Release 4 integriert zwar viele der oben erwähnten Features, manche aber nur in Kompatibilitätslibraries wie das ganze Socket-Interface. TLI bleibt auch weiterhin das in der System V Interface Definition festgelegte Netzwerk-Interface. So sollen Programmierer ermuntert werden, das TLI anstatt der Sockets zu benutzen.

Einheitlicher Systemzugang

Wie schon beschrieben, bietet Unix einige Möglichkeiten, Verbindungen zu anderen Rechnern aufzubauen, sei es über Netzwerke oder auch nur über serielle Leitungen. Wie sieht es aber im umgekehrten Fall aus? Welche Möglichkeiten bietet Release 4 an, um Kommunikationsanforderungen zu erkennen, insbesondere den Wunsch, Zugang zum System zu erhalten? Bisher wurde hierbei stark nach der Art der Verbindung unterschieden. Beim Einloggen über eine serielle Leitung wurde jeweils ein *getty*-Prozeß für jede *tty*-Line gestartet, das BSD-Unix startet

	Pre-SVR4 UNIX System History	System V Release 4	SVR4 Enhanced Security	SVR4 Multiprocessing Plus	SVR4 Network Computing Plus
Internationalization	8-bit Character Support National systems	Multi-National Language Support	Added multi-language and GUI support	Enhanced Multi-byte Character Support	
Systems Administration	Basic Operations Administration and Maintenance				
Multiprocessing					
Network Computing					
Security					
User Interface Functions					
Core Operating System					

Standards Conformance: XPG3, POSIX 1003.1 XPG4, POSIX 1003.2 XPGn, POSIX 1003...

Langjährige Planungen: Die System V Roadmap

Quelle: Unix International

einen Port-Monitor für Netzwerkverbindungen.

Um nicht zwischen den verschiedenen Arten des Systemzuganges unterscheiden zu müssen, führt Release 4 die 'Service Access Facility (SAF)' ein. Dies ist eine Schicht, die über den einzelnen Port-Monitoren angesiedelt ist und so einen einheitlichen Zugriff auf alle möglichen Systemzugänge bietet. Dadurch kann man von den jeweils zugrundeliegenden Verbindungen abstrahieren und diese gemeinsam administrieren, und zwar mit dem *sacadm*-Kommando, das dem Systemverwalter erlaubt, Port-Monitore zu konfigurieren.

Release 4 enthält drei verschiedene Port-Monitore, *listen* und *inetd* sowie *trymon*. Der Internet Superserver *inetd* ist ein Port-Monitor für TCP/IP-Netzwerke. Während der *inetd* nur für eine bestimmte Protokollfamilie gedacht ist, bietet der *listen*-Dämon einen verbindungsorientierten, protokollunabhängigen Service. Der *trymon* ersetzt das aus früheren Unix-Versionen bekannte *getty*. Mit dem *trymon* können mehrere serielle Ports beobachtet werden. Es ist nicht wie beim *getty* erforderlich, für jeden Port einen Prozeß zu starten.

Standard-Signale

Auch im Bereich des Prozeß-Managements integriert Release 4 viele Funktionen aus dem Berkeley-Unix. Mit Hilfe der Job Control (optional in POSIX P1003.1 festgelegt) kann der Benutzer Prozesse in den Hintergrund legen, in den Vordergrund holen, stoppen oder auch wieder aktivieren, je nachdem, welche Prioritäten er ihnen zubilligt. Solche Features wurden in der Job Control Shell (*jsh*) implementiert. Dieses ist auch eine Anforderung an die in Unix verwendeten Signale. Das POSIX-Signal-Interface bietet neben der Job Control-Unterstützung Funktionen, um eine bestimmte Menge von Signalen zu manipulieren, Signale zu maskieren und zu demaskieren sowie die Möglichkeit, bereits wartende Signale zu untersuchen. Dies ist das in Release 4 verwendete Standard-Signal-Interface, daneben werden aus Kompatibilitätsgründen noch die alten Signal-

System-V-4.0-Kommandos aus BSD oder Xenix

Hinzugekommene BSD-Kommandos

<i>arx</i> (1M)	<i>last</i> (1)	<i>rsh</i> (1)
<i>atd</i> (1)	<i>more</i> (1)	<i>rshd</i> (1M)
<i>atrm</i> (1)	<i>mountd</i> (1M)	<i>ruptime</i> (1)
<i>automount</i> (1M)	<i>named</i> (1M)	<i>rusers</i> (1)
<i>biod</i> (1M)	<i>netstat</i> (1M)	<i>rusersd</i> (1M)
<i>bootparamd</i> (1M)	<i>nfsd</i> (1M)	<i>rwall</i> (1)
<i>chkey</i> (1)	<i>nsllookup</i> (1M)	<i>rwalld</i> (1M)
<i>clear</i> (1)	<i>page</i> (1)	<i>rwho</i> (1)
<i>compress</i> (1)	<i>ping</i> (1)	<i>rwhod</i> (1M)
<i>csn</i> (1)	<i>quot</i> (1M)	<i>script</i> (1)
<i>ctags</i> (1)	<i>quota</i> (1M)	<i>showmount</i> (1M)
<i>dump</i> (1)	<i>quotacheck</i> (1M)	<i>spray</i> (1)
<i>dumpps</i> (1M)	<i>quotaoff</i> (1M)	<i>sprayd</i> (1M)
<i>edquota</i> (1M)	<i>quotaon</i> (1M)	<i>statd</i> (1M)
<i>exportfs</i> (1M)	<i>rarpd</i> (1M)	<i>strings</i> (1)
<i>finger</i> (1)	<i>rcp</i> (1)	<i>telnet</i> (1M)
<i>fingerd</i> (1M)	<i>rcpbind</i> (1M)	<i>telnetd</i> (1M)
<i>fmt</i> (1)	<i>rdump</i> (1M)	<i>tfpd</i> (1)
<i>fdt</i> (1)	<i>requota</i> [ufs]dumb(1M)]	<i>tfpd</i> (1M)
<i>fdx</i> (1)	<i>restore</i> (1M)	<i>timed</i> (1M)
<i>tpd</i> (1M)	<i>rexeod</i> (1M)	<i>trpt</i> (1M)
<i>gcore</i> (1)	<i>rksh</i> [ksh](1)]	<i>tunefs</i> (1M)
<i>gettable</i> (1M)	<i>rlogin</i> (1)	<i>uncompress</i> (1)
<i>head</i> (1)	<i>rlogind</i> (1M)	<i>uuencode</i> (1C)
<i>htable</i> (1M)	<i>route</i> (1M)	<i>uuencode</i> (1C)
<i>lccnfig</i> (1M)	<i>routed</i> (1M)	<i>whos</i> (1)
<i>inetd</i> (1M)	<i>rpgen</i> (1)	<i>zcat</i> (1)
<i>keyenvoy</i> (1M)	<i>rcpinfo</i> (1M)	<i>zdump</i> (1M)
<i>keyserv</i> (1M)	<i>rquotad</i> (1M)	<i>zic</i> (1M)
<i>ksh</i> (1)	<i>rrestore</i> [ufs]restore(1M)	

Zusammengefaßte System-V-/BSD-Kommandos

<i>admin</i> (1)	<i>fsck</i> (1M)	<i>rmdel</i> (1)
<i>at</i> (1)	<i>get</i> (1)	<i>sccsdiff</i> (1)
<i>bci</i> (1)	<i>id</i> (1M)	<i>sed</i> (1)
<i>chgrp</i> (1)	<i>initt</i> (1M)	<i>sh</i> (1)
<i>chmod</i> (1)	<i>join</i> (1)	<i>sort</i> (1)
<i>chown</i> (1)	<i>kill</i> (1)	<i>spline</i> (1G)
<i>cp</i> (1)	<i>ln</i> (1)	<i>split</i> (1)
<i>cpio</i> (1)	<i>login</i> (1)	<i>stty</i> (1)
<i>crontab</i> (1)	<i>ls</i> (1)	<i>tail</i> (1)
<i>date</i> (1)	<i>mach</i> [uname](1)]	<i>tar</i> (1)
<i>dc</i> (1)	<i>mkfs</i> (1M)	<i>test</i> (1)
<i>deltal</i> (1)	<i>mount</i> (1M)	<i>touch</i> (1)
<i>devnm</i> (1M)	<i>mv</i> (1)	<i>umount</i> (1M)
<i>df</i> (1M)	<i>ncheck</i> (1M)	<i>val</i> (1)
<i>diff</i> (1)	<i>od</i> (1)	<i>vi</i> (1)
<i>diff3</i> (1)	<i>pr</i> (1)	<i>vmstat</i> [sar](1)]
<i>echo</i> (1)	<i>prn</i> (1)	<i>w</i> [whodo](1M)]
<i>ed</i> (1)	<i>pstat</i> [swap](1M)]	<i>wall</i> (1M)
<i>find</i> (1)	<i>rm</i> (1)	<i>write</i> (1)

Zusammengefaßte Xenix-/System-V-/BSD-Kommandos

<i>crash</i> (1M)	<i>grep</i> (1)	<i>passwd</i> (1)
<i>cron</i> (1M)	<i>initt</i> (1M)	<i>su</i> (1M)
<i>echo</i> (1)	<i>ipcs</i> (1M)	<i>sulogin</i> (1)
<i>file</i> (1)	<i>login</i> (1)	<i>tar</i> (1)
<i>fsck</i> (1M)	<i>ls</i> (1)	
<i>fsdb</i> (1M)	<i>mknod</i> (1M)	

Quelle: [1]

schnittstellen angeboten: *signal* aus den Releases 2.x sowie *sigset* aus Release 3.0.

Begrenzt echtzeitfähig

Unix ist an sich als Time-Sharing-System konzipiert – alle lauffähigen Prozesse bewerben sich um die CPU. Nach bestimmten Kriterien wird einer

davon ausgewählt und bekommt für einige Zeit die CPU zugeteilt. Danach wird er gestoppt, um eventuell einem anderen Prozeß die CPU zu geben. Die Auswahl nimmt der Scheduler vor. In Release 4 gibt es spezielle Schedulingklassen (man kann auch eigene definieren), jeweils eine für Time-Sharing, Kernel-Prozesse sowie

Real-Time-Jobs, die eine hohe Priorität aus der Kernel selbst haben. Mit *dispaamin* kann man die Scheduling-Politik weiter beeinflussen. So lassen sich die maximale Zeitscheibe, die ein Prozeß erhält, und Prioritätenänderungen bei Zustandswechseln einstellen. Dadurch ist das Scheduling-Verhalten des Systems konfigurierbar geworden. Das eröffnet jedoch auch neue Probleme. Durch ungünstiges Konfigurieren kann die Gesamtleistung des Systems stark abfallen, ein Real-Time-Prozeß sogar alle anderen Prozesse (schlimmstenfalls die des Kernels) blockieren. Aus diesem Grunde darf nur *root* solche Prozesse aufsetzen.

Mit Hinblick auf das Debugging von Prozessen gibt es in Release 4 das *proc*-Filesystem. Dieses Filesystem ermöglicht es, mit Prozessen wie mit normalen Files zu arbeiten: *open*, *close*, *read*, *write* und *ioctl* sind auch für Filesystem-Typen wie *proc* definiert. Das erleichtert die Fehlersuche in Prozessen. Beispiele für Debugging-Anwendungen sind die Kommandos *truss* und *gcore*. Das Kommando *truss* kann (auch bereits laufende) Prozesse tracen, die aufgerufenen Systemcalls werden samt Argumenten angezeigt. So etwas ist auch mit *traceptrace* möglich, aber bei weitem nicht so komfortabel und einfach. Der Befehl *gcore* schreibt das Core-Image des Prozesses in ein File, das der Debugger *sdb* lesen kann.

AT&T hat auch an die Zukunft ihres Kindes gedacht. Release 4.0 ist vorbereitet für den Einsatz auf Multiprozessor-Systemen – der wachsende Bedarf nach steigender Rechenleistung ist absehbar. Übrigens auch ein steigender Massenspeicherbedarf – für System V 4.0 sollte man schon 100 MByte Plattenkapazität einplanen, zuzüglich natürlich der Benutzerdaten.

On the road again

Die Roadmap beschreibt die geplante Entwicklung des Unix System V Release 4 unter der Regie von AT&Ts Unix Software Operation (siehe Bild). Obwohl AT&T ihr Release 4 schon ausliefert, wird es noch einige Jahre dauern, bis

✓

✓

✓

✓

die volle Funktionalität erreicht ist. Der zeitliche Rahmen erstreckt sich hier auf vier bis fünf Jahre. Erweiterte Security-Features wie das Abweisen unzulässiger Zugriffe von Personen oder Viren durch Multi-Level-Integrität und Auditing-Funktionen, Vermeidung von Datenkorruptionen sowie eine angestrebte B2-Klassifizierung (und höher) werden erst für das erste Quartal 1991 erwartet. Bis zum Sommer 1992 will man die Multiprozessor-Unterstützungen einarbeiten, die mit Highlights wie symmetrischem Shared Memory und einem standard multiprocessing applications programmer interface (API) aufwarten wollen.

Es geht dabei um Folgendes: Bei Multiprozessor-Architekturen gibt es mehrere Möglichkeiten, die einzelnen Prozessoren untereinander und mit dem Speicher zu verbinden. Jeder Prozessor kann seinen eigenen lokalen Speicher haben, oder mehrere CPUs teilen sich einen Speicher. Natürlich gibt es nicht nur diese beiden Extreme (alle oder ein Prozessor pro Speicher), sondern auch Kombinationen. Prozessoren können über einen kleinen lokalen Speicher verfügen und trotzdem Zugriff auf 'weiter entfernten' haben. Können alle Prozessoren gleichberechtigt auf den Speicher zugreifen, spricht man von einem 'symmetrischen Shared Memory'.

Ein Schritt zur Einheit

Dieses zum Thema Multiprocessing. Dem Bereich Netzwerkerweiterungen mißt man anscheinend nicht so viel Bedeutung bei. Die sollen erst zwischen Ende 1993 und Sommer 1994 eingeführt werden. Geplant sind Unterstützung von OSI-Protokollen und verteilten Applikationen sowie die Zusammenarbeit mit anderen Netzwerksystemen. Auch für die System- und Netzwerkverwaltung stehen Verbesserungen an. Dazu zählt AT&T Features wie Auto-Konfiguration, bessere Performance-Analyse des Systems sowie bessere Unterstützung bei heterogenen Netzen. Diese Entwicklung wird stark vorangetrieben, wie der erwartete Einsatz von rund 100 Millionen Dollar zeigt.

Alles in allem bietet das neue System V Release 4 für Unix-Programmierer aller Couleur eine gute Arbeitsgrundlage. Wer jedoch BSD- oder SYSV-spezifisch programmieren will, kann dies auch weiterhin tun. Wenn sich der Programmierer nicht an die ihm zur Verfügung stehenden Hilfsmittel (ABI et cetera) hält, kann er oder sie genauso unportable Programme schreiben wie vorher. Denn leider läßt sich nicht alles so elegant unter einen Hut bringen wie die verschiedenen Filesysteme.

Der erste Schritt zu einem einheitlichen Unix ist aber getan. Jetzt sind die Hersteller, die an der Realisierung mitgewirkt haben, weiter gefordert. Auf lange Sicht läßt sich das praktizierte Schema von 'vier in eins' nicht durchhalten, wenn sich die einzelnen Hersteller nicht darum bemühen, einen einheitlichen Standard zu schaffen. In bestimmter Hinsicht ist erst Release 4.1 Prüfstein für die Ernsthaftigkeit der Standardisierungsbestrebungen – an dessen technischen Spezifikationen wirken nämlich die Mitglieder von Unix International mit.

Vielleicht erweist sich der Ansatz, den die CMU (Carnegie Mellon University) mit ihrem Mach-Betriebssystemkern gewählt hat, in Zukunft als flexibler, um auf den Bedarf des Marktes einzugehen. Hat man erst einmal den Mach-Kernel auf eine Maschine portiert, so ist es relativ einfach, das gerade passende Interface, sei es nun BSD, System V oder eine Eigenentwicklung, auf diesem Kern aufzusetzen. Dies hat offenbar auch die OSF erkannt, die Mach zu ihrem 'Betriebssystem der Zukunft' erkoren hat. (JS)

Jörg Hattenhauer und Michael Kienle arbeiten am FB Informatik der Universität Dortmund und beschäftigen sich mit Unix und lokalen Netzen.

Literatur

- [1] Unix System V Release 4 Migration Guide
- [2] POSIX Norm 1003.1
- [3] Uwe Krüger, UNIX – Legende mit Zukunft, iX 2 und 3/89
- [4] Torsten Beyer, Michael Kuschke, Begegnungen der vierten Art, iX 2/89



Je besser die Organisation, um so größer der Erfolg.

Wir machen UNIX -

- mit innovativen Produkten
- mit Anwendungen, die man braucht
- mit Netzwerken, die verbinden
- mit Werkzeugen, die helfen
- mit leistungsstarker Hardware
- mit effizienter Schulung
- mit Rat, und Tat, und vielem mehr.

Heute präsentieren wir Ihnen:

UNIPLEX Business Software. Je besser die Organisation, um so größer der Erfolg.

UNIPLEX, das ist die ausbaufähige Business Software für Ihr Büro. Von UNIPLEX, der No.1 auf dem Markt der modernen, integrierten Burosysteme. Mit Textverarbeitung, Datenbank, Tabellenkalkulation, Terminkalender, Elektronischer Post und Grafik in einzigen System. Und mit offenen Schnittstellen zu UNIX, INFORMIX und ORACLE, die die Integration weiterer Anwendungen unterstützt. Wenn Sie mehr über UNIPLEX Business Software wissen wollen, wenden Sie sich an unseren Distributor WORKSTATION 2000.

UNIPLEX

Wenn Sie Ihre Software optimal nutzen möchten, empfehlen wir Ihnen als Hardware die Power Serie RISC/6000 von IBM.

WORKSTATION 2000 GmbH
Baierbrunner Str. 31, 8000 München 70
Tel.: 78 80 34, Fax: 780 97 92

Wir machen UNIX.

✓

✓

✓

✓

3

Kapitel 3

Kapitel 3 : Installation

Neuinstallation : Übersicht.....:	III-1-10
Installation von SW-Produkten ...:	III-2-10
COLLAGE-Bediensystem	III-3-10

Installation SINIX V5.4

Übersicht : Neuinstallation SINIX V5.4

SINIX0 : SINIX0 einlegen + MX300 einschalten
Aufforderung : SINIX1

SINIX1 : Kernel wird geladen
File-System s5 FS ins Ram-Disk

INSTALL

Aufforderung : SINIX2

SINIX2 : mount /dev/fd0 /install
File-System s5 Tasterabfrage

INSTALLA

Platteneinteilung : fdisk (partitions)
disksetup (slices)

Minisystem wird auf Platte installiert
disklabel
reboot

Platte (Mini) : Platten-Kernel wird geladen

INSTALL2

Aufforderung : SINIX3

SINIX3 : root-FS wird von Tape auf Platte kopiert

Archiv 1 : Header
Archiv 2 : root-FS
Archiv 3 : Packages

INSTALL3

additional Filesysteme werden eingerichtet
Passwörter eingeben : root, install, tele
Rechnername eingeben

INSTALL4

Foundation Set einspielen (pkgadd all)
Passwörter eingeben : admin, sysadmin
Foundation Set ist eingespielt
Unix wird neu gebunden

Installation komplett
reboot

Platte (Normal): Key-Diskette einlegen

Multi-User-Mode

COMANDI X INSTALLARE SOFTWARE

Installation und Deinstallation von Softwareprodukten

Zur Installation, Deinstallation, sowie Information über installierte Softwareprodukte stehen Ihnen die Kommandos

Si lavora da # (shell) superuser

quasi = excl

pkgadd = x *installare pacchetti standard effect. software*
(non su più linee)

pkgchk = x *verificare correttezza*

pkginfo = *quasi* *installare*

pkgparam =

pkgrm = x *disinstallare*

zur Verfügung.

1) pkgadd (transfer Software package to the system)

Mit Hilfe dieses Kommandos können Sie (*nachträglich*) Softwareprodukte auf Ihrer MX300I installieren.

Beispiel: Nachträgliches Installieren eines Softwarepakets vom Installationsband.

/usr/sbin/pkgadd -d ctape1 = x *STREM*.
OPZIONE
DISKETTE 1 = x FD 3 1/2

Beispiel: Installation eines/mehrerer Softwarepakete von einer 3 1/2 Zoll Floppy.

/usr/sbin/pkgadd -d diskette1

HINWEIS: Sollte es während der Installation eines Paketes zu Fehlern kommen (Meldung: package installation failed), so sind Teile des "packages", die vor der Meldung eingelesen wurden, schon installiert. In diesem Fall ist es empfehlenswert das Paket mittels dem Kommandos pkgrm explizit wieder zu entfernen, bevor Sie dieses Paket erneut installieren.

2) pkgchk (check accuracy of installation)

Mit diesem Kommando können Sie überprüfen, ob das angegebene Paket korrekt installiert wurde, bzw. mit dem Schalter "-l" können Sie sich detaillierte Informationen über ein Paket ausgeben lassen.

Beispiel: Informationen über das Paket "oam"

```
/usr/sbin/pkgchk -l oam
```

3) pkginfo (display Software package information)

An dieser Stelle soll nur auf den Schalter "-l" des Kommandos eingegangen werden. Mit Hilfe dieses Schalters erhalten Sie alle verfügbaren Informationen der installierten Pakete.

Beispiel:

```
/usr/bin/pkginfo -l
```

liefert unter anderem folgende Informationen über jedes installierte Paket:

PKGINST	Kurzname des Pakets
NAME	Erläuterung des Paketinhalts
CATEGORY	Paketzugehörigkeit (system für ST-Produkte, application für andere)
ARCH	Systemarchitektur (i386)
VERSION	Versionsnummer
VENDOR	wer über das COPYRIGHT verfügt
HOTLINE	Telefonnummer einer Ansprechstelle
INSTDATE	Installationsdatum
STATUS	komplett/teilweise installiert
FILES	-----> x installed pathnames
	----> x shared pathnames
	----> x linked files
	----> x directories
	----> x executables
	----> x setuid/setgid executables
	----> x block used (approx)

4) pkgparam (displays package parameter values)

Mit Hilfe dieses Kommandos werden alle mit einem Paket assoziierten Parameter aufgelistet.

Beispiel:

```
/usr/sbin/pkgparam -v ces
```

liefert alle assoziierten Parameter des ces-packages.

5) pkgrm (removes a package from the system)

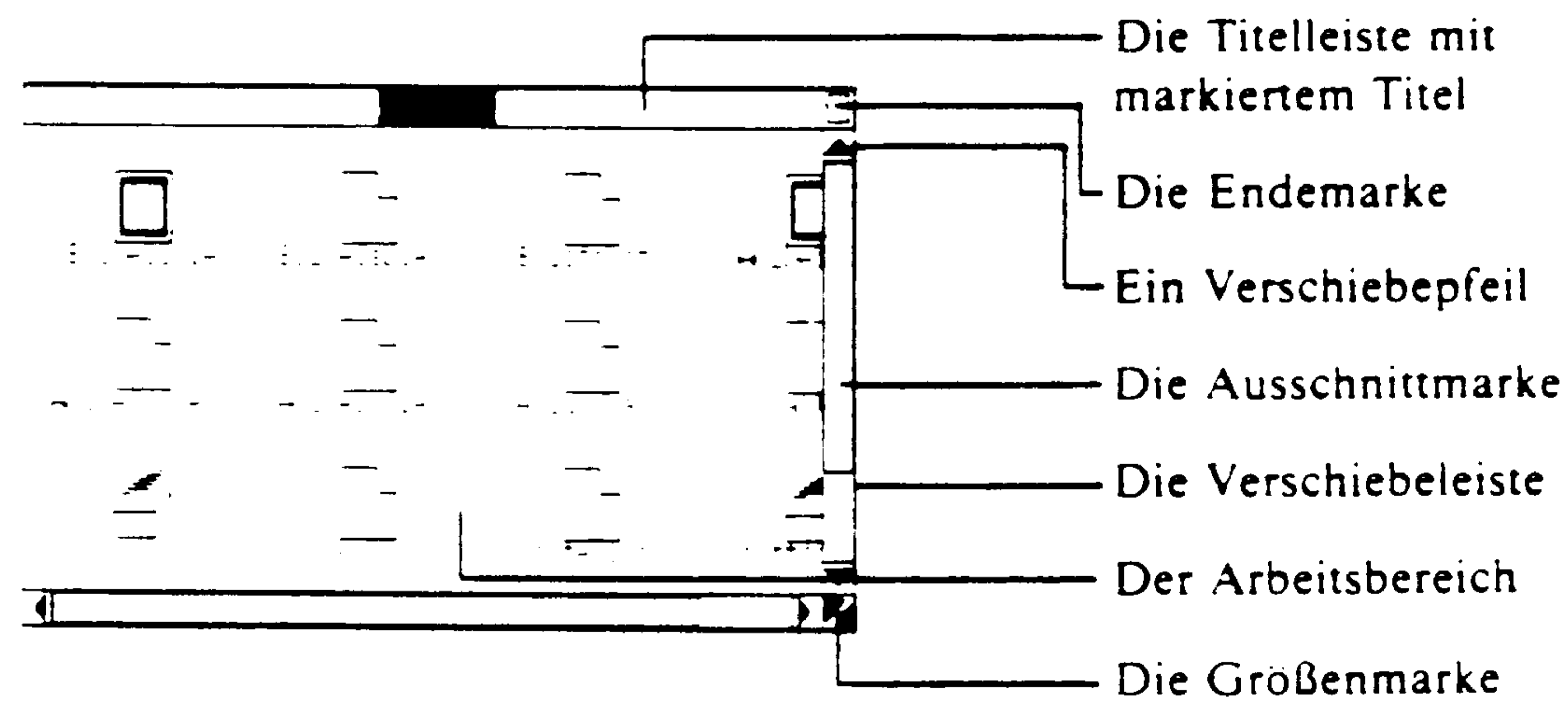
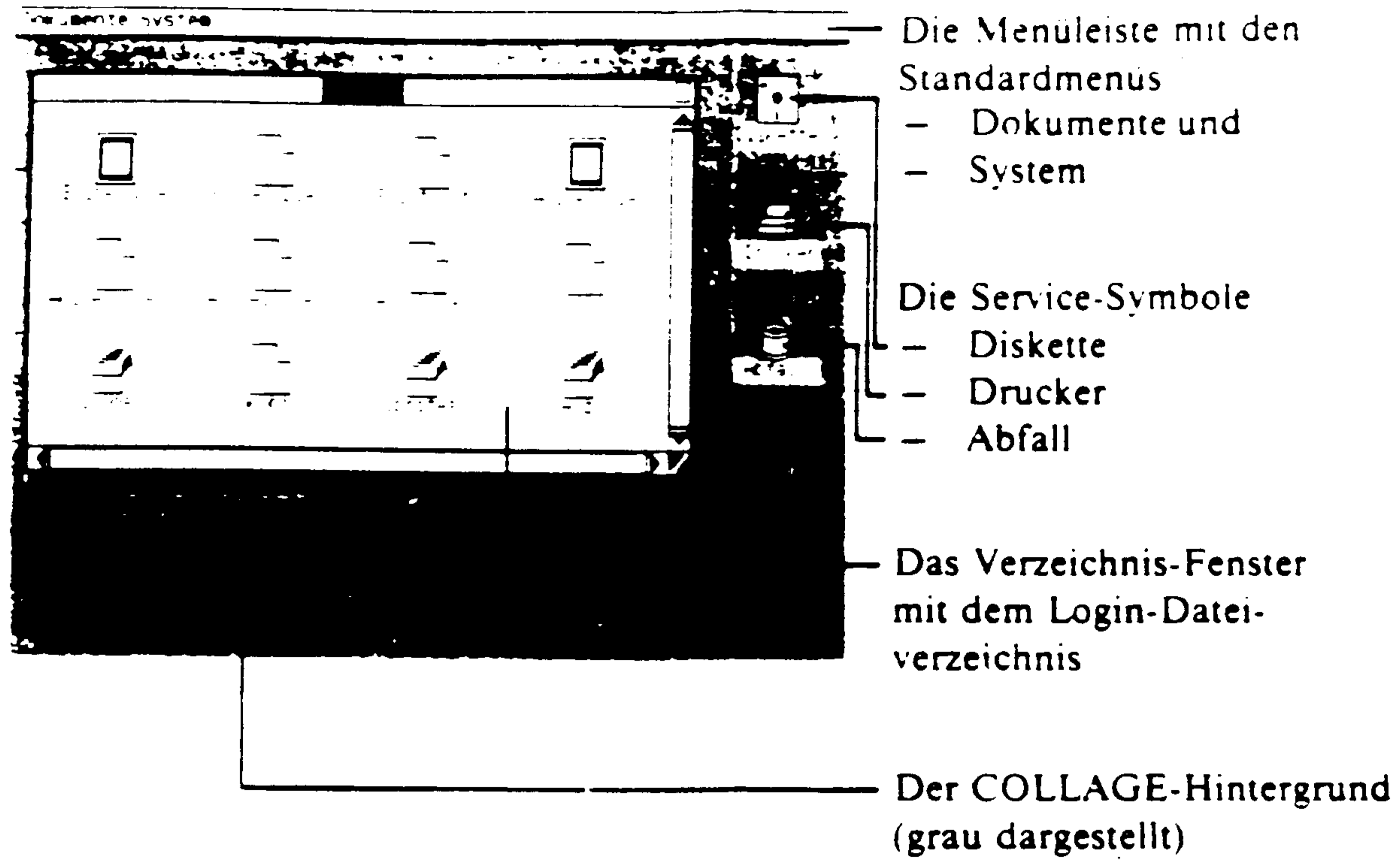
Durch den Aufruf des Kommandos

```
/usr/sbin/pkgrm
```

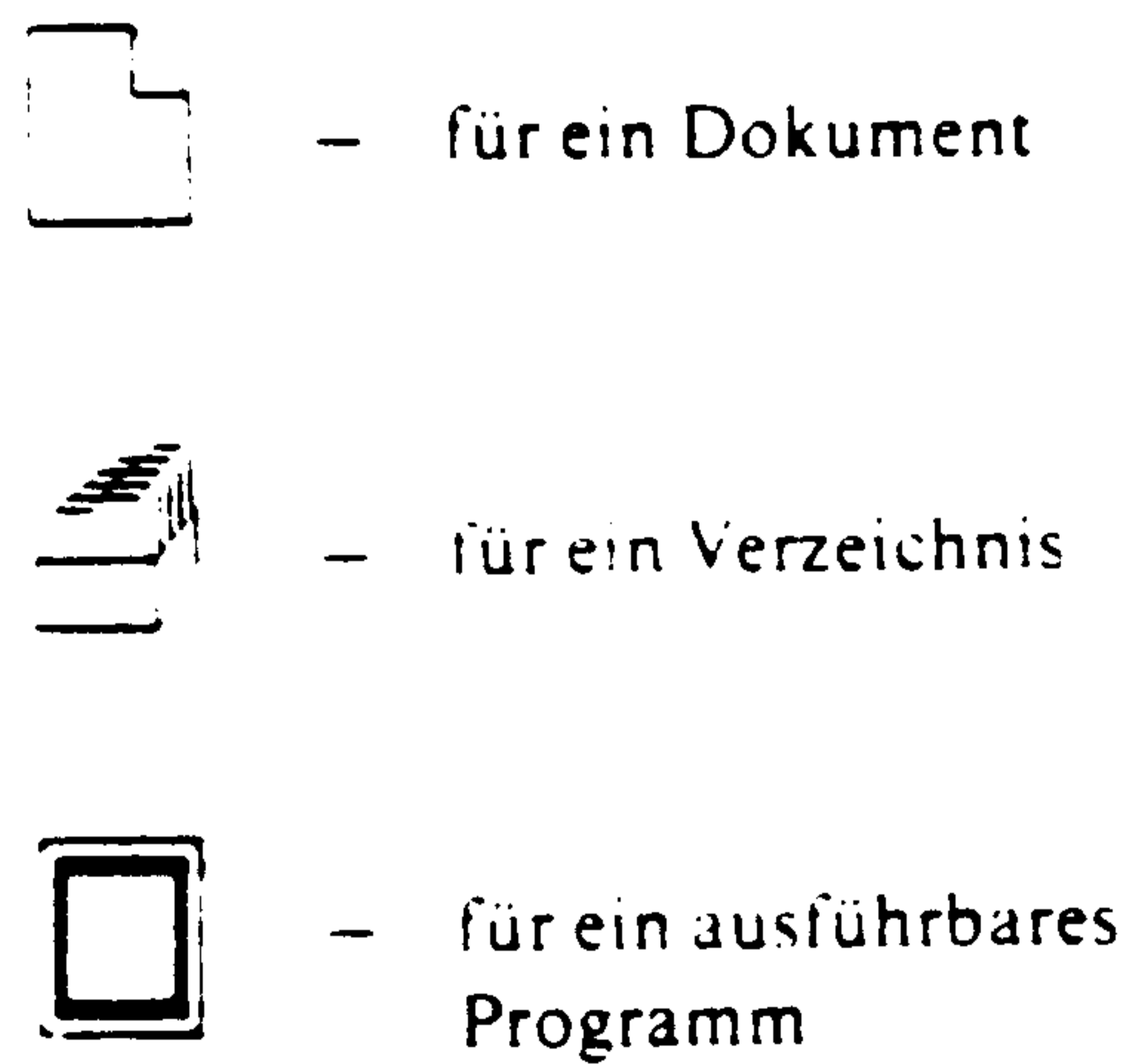
erhalten Sie eine Übersicht der installierten Softwarepakete. Sie können hierbei menügeführt die Pakete selektieren, die Sie deinstallieren möchten.

COLLAGE-Bedienung

Collage Bedieneroberflaeche



Die Standard-Symbole




Alpha-COLLAGE : Bedienung

Deutsche Sprachoberfläche: LANG=De_DE.o4b

Aufruf: collage
oder acollage

Oberfläche: Desktop
Verzeichnis Ablage
Icons

Maus: Aktivieren mittels MODE-Taste: **+** = aktiv
█ = inaktiv
Bedienung wahlweise: - Cursor-Tasten,
RETURN-, START-Taste, ,
oder:
- Funktions-Tasten (Maske!)

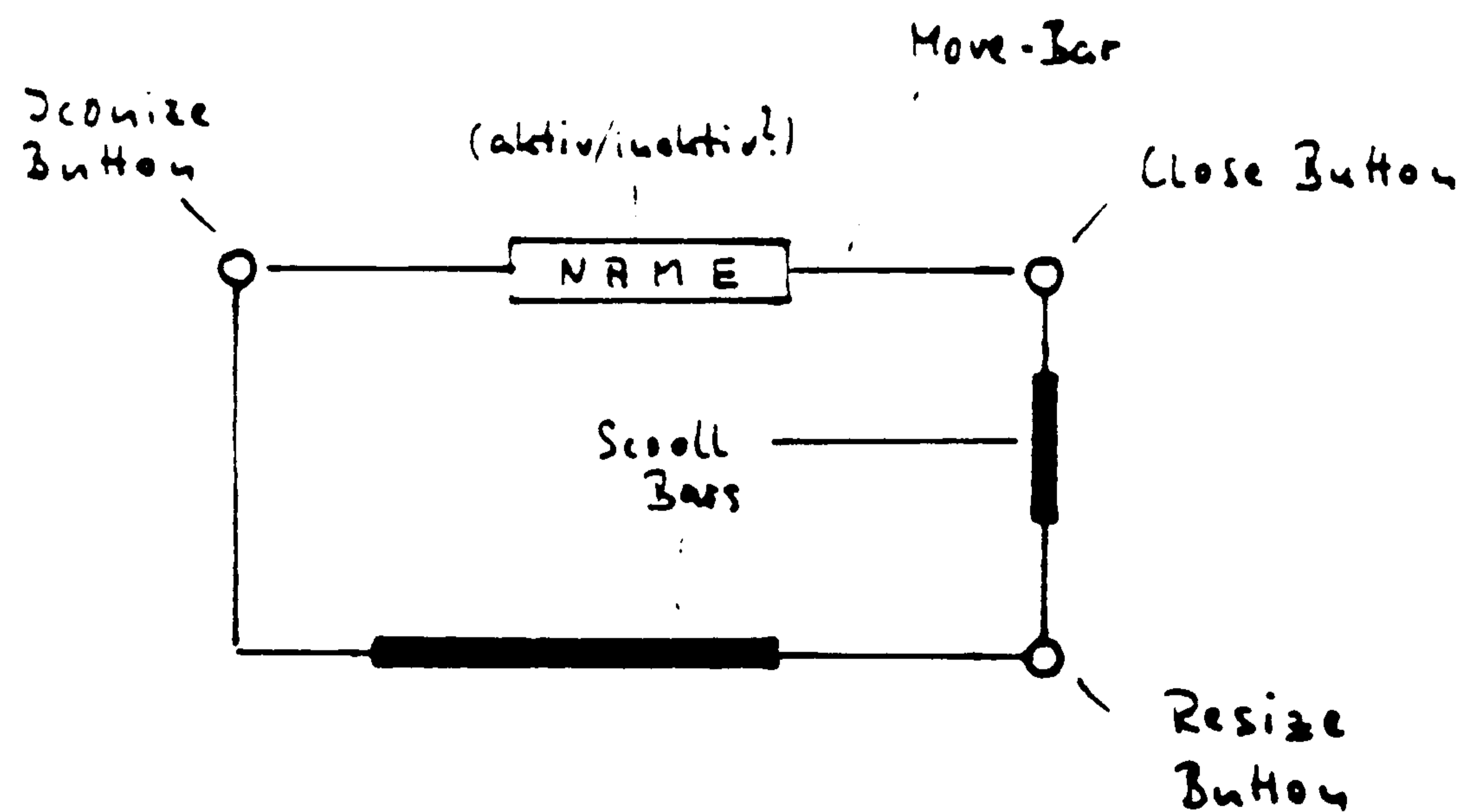
Pulldown-Menüs:

Auswahl: MENU-Taste, Cursor-tasten

Aktivieren: RETURN-Taste

Menüs: - Desktop: Immer sichtbar,
 betrifft alles außerhalb von Fenstern
- Bearbeitung,
 Ablage: Nur sichtbar bei aktivem Verzeichnis-
 fenster
- Standard: "Oldstyle-Fenster"

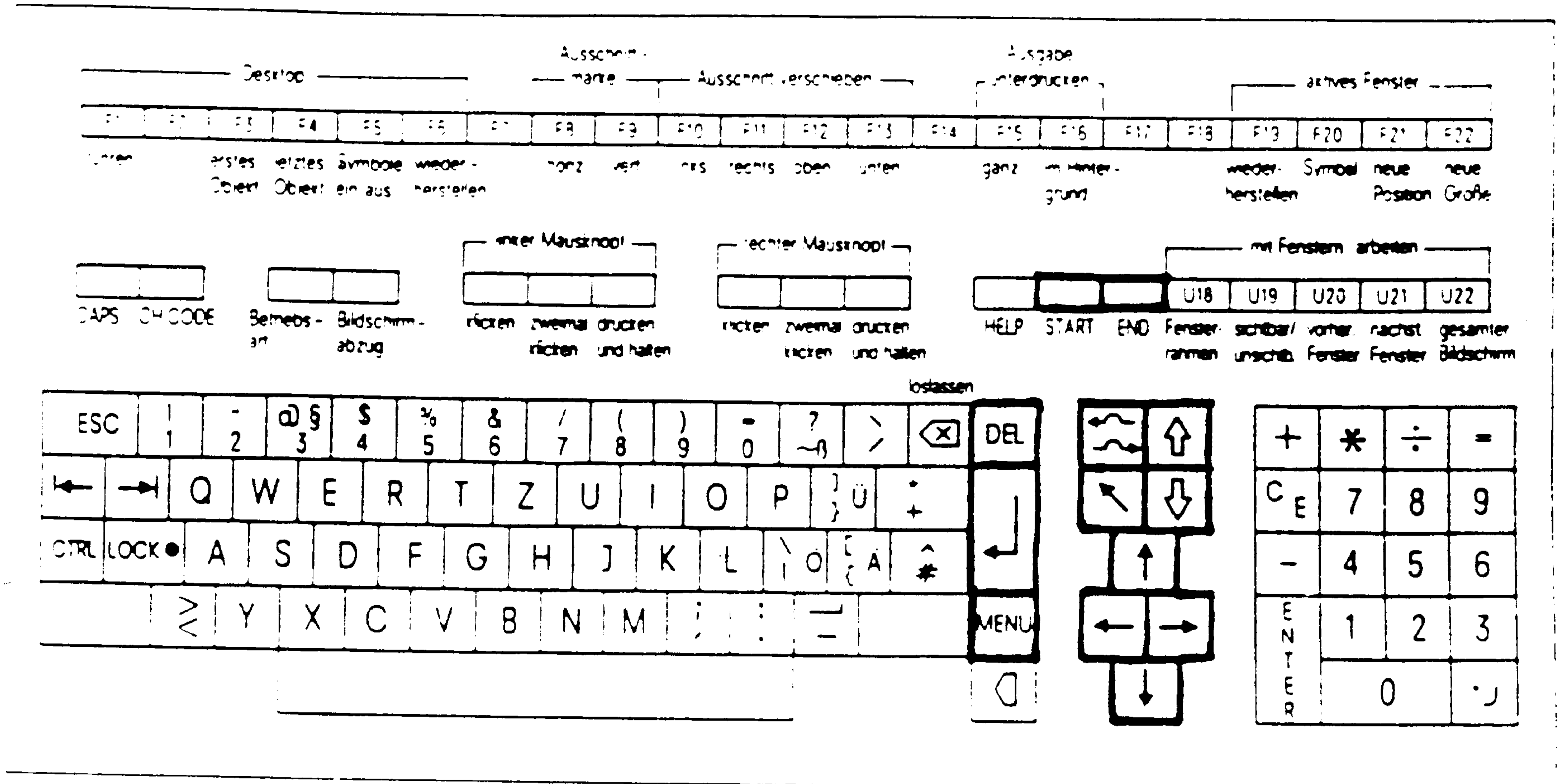
Fenster:



ACHTUNG !

Damit an Oldstyle-Applikationen Menü- und funktionstasten durchgereicht werden, muß im Standardmenü die Funktion *U-Tasten* übergeben aktiviert werden.

Wichtige Tasten in alphanumerischer Umgebung



MENU

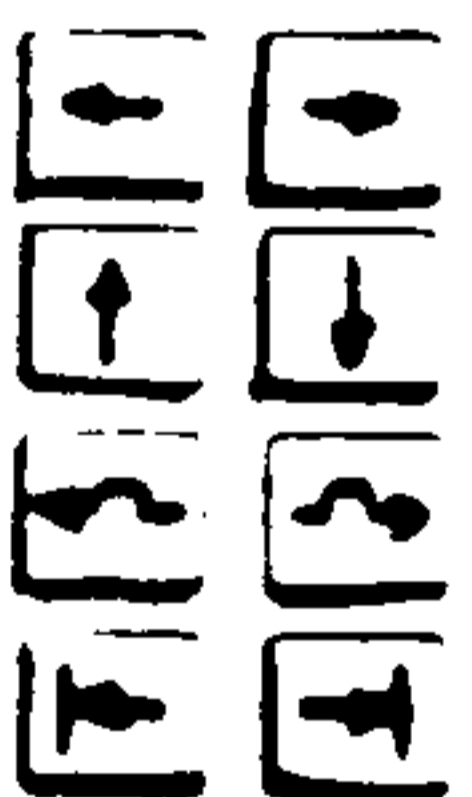
Ein Menü ausbreiten.



Zwischen Menüs in der Menüleiste wechseln.



- Innerhalb eines Menüs zwischen Befehlen wechseln.
- Innerhalb einer Liste zwischen Einträgen wechseln.



Innerhalb eines Verzeichnis-Fensters zwischen Objekten wechseln.



Den Zeiger auf das erste Objekt bringen.



- Ein Objekt in einem Verzeichnis-Fenster markieren.

DEL

- Auslösen von Quittierfeldern, die mit 'DEL' beschriftet sind.

START

- Starten eines markierten Service-Programmes.

MENUS

Desktop

- Anwendungen
 - Shell
 - Bildschirmabzug
- Blättern
- Verzeichnis
- Bildschirm
- Aktualisieren
- Information
- Löschen
- Umbenennen
- Beenden

Bearbeitung

- Offnen
 - Editieren
 - Lesen
 - Seitenweise lesen
- Information
- Erzeugen
- Löschen
- Auswählen
- Kopieren
- Ausschneiden
- Duplizieren
- Umbenennen

Ablage

- Schließen
- Darstellung
- Einstellung
- Änderung
- Aktualisieren

SERVICE-SYMBOLS

Diskette

- Service-Programm
 - Diskette
 - Inhaltsverzeichnis lesen
 - Alles lesen
 - Eigentümer ändern
 - Formatieren

Kassette

- Service-Programm
 - Kassette
 - Inhaltsverzeichnis lesen
 - Alles lesen
 - Formatieren

Post

- Menu
 - 'Post'
 - Bearbeiten
 - Senden
 - Umlenken
 - Beenden

Standard

- Zwischenspeicher
- Kopieren
- Holen
- ISAR12
- ISAR16
- CHCODE
- U-Tasten abfangen (Alpha)
- U-Tasten übergeben (Alpha)

Dienste+Info

- Kennwort ändern
- Sprache einstellen
- Benutzerinfo
 - Alle Benutzer
 - Aktive Benutzer
- Plattenbelegung
- Aktive Prozesse
- Konfigurierung zeigen
 - Menü 'Konfigurierung'
 - Board anzeigen
 - AK-Namen anzeigen
 - Beenden
- Installierte Software
- Postzustand

Entwicklung

enthält die Namen von installierten Entwicklungs- und Programmiersystemen

Netze

enthält die Namen von installierten Kommunikationsprogrammen

Drucker

Verwaltungs-Programm

- 'Drucker'
 - Menü 'Druckaufträge'
 - Ändern
 - Druckergruppe ändern
 - Priorität ändern
 - Löschen
 - Voreinstellen
 - Menü 'Einstellungen'
 - Drucker
 - Menü 'Drucker' (nur für Drucker- und Systemverwalter)
 - Sperrern
 - Freigeben
 - Testen
 - Vordruck setzen
 - Voreinstellen (nur für Systemverwalter)
 - Druckergruppen
 - Druckerverwalter
 - Benutzer
 - Bildschirme
 - Vordrucke

Abfall

- Druck-Programm
 - Druckauftrag erteilen

Menü 'Systemverwaltung'

- Konfigurierung
- Benutzerverwaltung
- Software-Installation
- Spoolverwaltung
- Postverwaltung
- Systemsprache

erhält nur der Systemverwalter

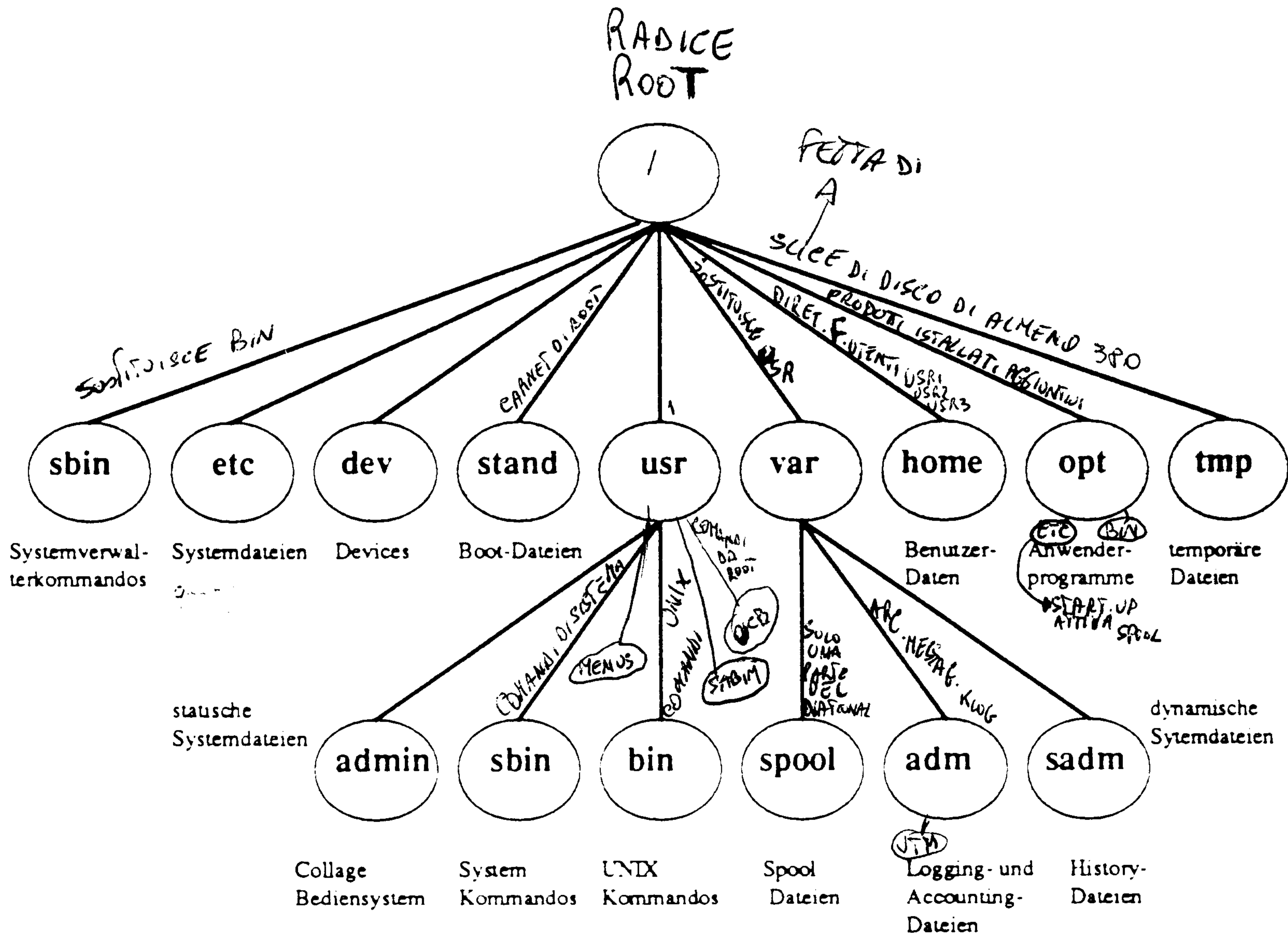
4

Kapitel 4

Kapitel 4 : SINIX V5.4

Filesystem	IV-1-10
Systemdateien, Kommandos	IV-2-10
Systemstart	IV-3-10
Filesystem-Typen	IV-4-10
Plattenorganisation	IV-5-10
Mini-System	IV-6-10
SPOOL V3.0	IV-7-10

Filesystem in SINIX V5.4



→ COME CANCELLARE E AZZERARE FILE TEMPORANEI
CLEANUP

- FILE DATI
Dateiverzeichnis
- FILE SIST.
Filesysteme

TYPE LS = PER CERCARE I COMANDI

STRUTTURA SISTEMA

4. Dateisysteme

4.1 "root"-Dateisystem

/		
bin	gelinkt auf /usr/bin	
dev	Geräte-dateien	
export	Default-Wurzel für exportierte Dateibäume	
etc	Maschinenspez. Dateien zum Hochfahren der MX300-I; boot	
home	Standard-Dateisystem für Benutzerverzeichnisse	
install	Defaultwert für Installationen	
lib	gelinkt auf /usr/lib	
lost+found	Das Dateisystem wird vom Kommando "fsck" genutzt	
mnt	Default für temporäres Mounten	
opt	Dateiverzeichnis für Anwenderprogramme	
proc	Prozeß Dateisystem (<i>cont. file man.</i>)	
sbin	Zentrale Programme für Administration und Operating <i>Programme d'administration</i>	
shlib	"shared library" <i>librerie x lib</i>	
stand	Defaultwerte für den Boot <i>file d'installation</i>	
tmp	Temporäre Dateien des Systems	
usr	Standard-Dateisystem für System-dateien, die sich nicht ändern. <i>FILE NAME MODIFIED BY (FISSE)</i>	
var	Standard-Dateisystem für System-dateien, die sich ändern und <i>FILE MODIFIED BY</i> Hersteller-spezifische Dateien.	

4.2 "/dev"-Dateiverzeichnis:

Das Dateisystem enthält alle im System notwendigen Geräte-dateien. Einige Gerätedateien sind in DVZ's zusammengefaßt.

/dev/	<i>SOTTODIRETORI</i>	
VP	Gerätedateien für Plattenspiegelsysteme <i>stunt. subson. esonni (L.H.K.)</i>	
cmx	Gerätedateien für CMX <i>prod. di Comumec. SIEM.</i>	
col	Gerätedateien für COLLAGE	
dspi	Gerätedateien für EXOS - LAN	
dsk	Gerätedateien für Platte und Floppy <i>DEVICE X FLOPPY DISK</i>	
fd	Gerätedateien für Datei Deskriptoren <i>DESCRIZ FILE</i>	
inet	Gerätedateien für Netzanschlüsse <i>COLLEC. RETE</i>	
pts	Gerätedateien für Pseudoterminal <i>TTYP PSEUDO TERMINALI</i>	
rdsk	Raw-Devices für Platte und Floppy <i>DEVICE X A. e FLOPP.</i>	
rmt	Gerätedateien für das Bandgerät <i>UNITA NASTRO</i>	
sim	Gerätedateien für Takti <i>X TAKSI</i>	

Unter /dev liegen noch weitere Gerätedateien.

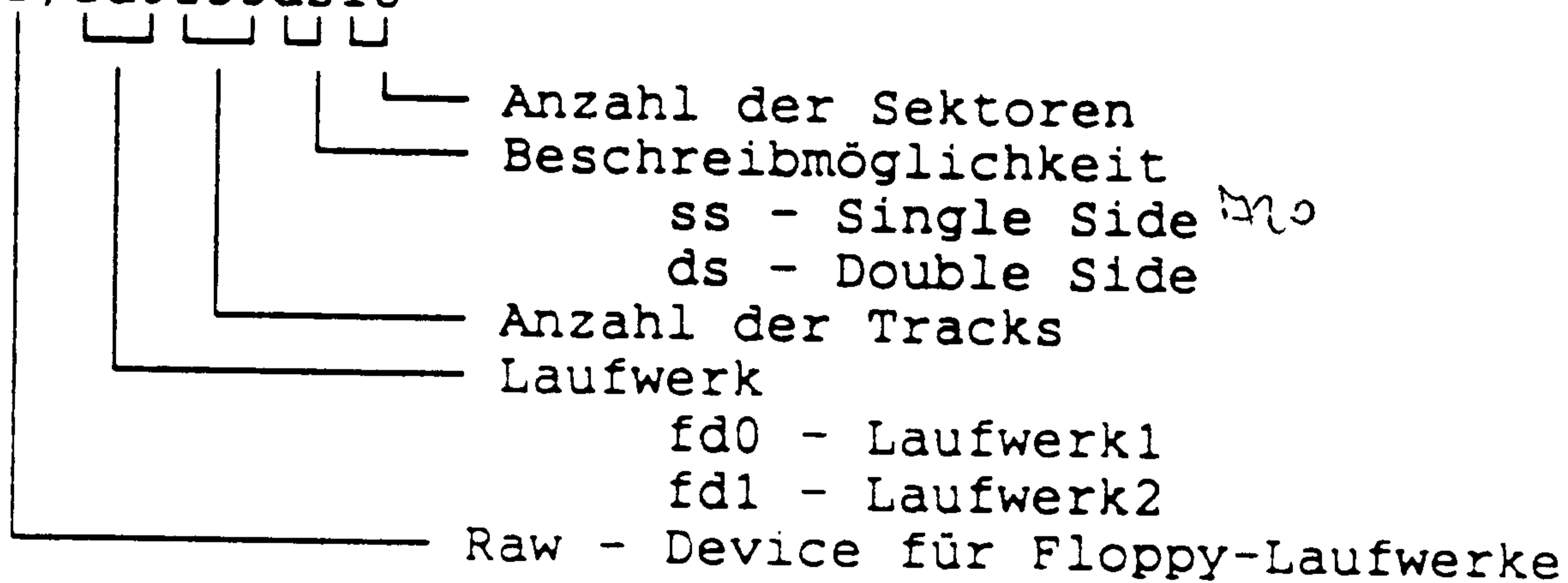
wie Macs ebenfalls, seine überlegen

exa0, exa0r, exa8 und exa8r für EXABYTE

Floppy Disk

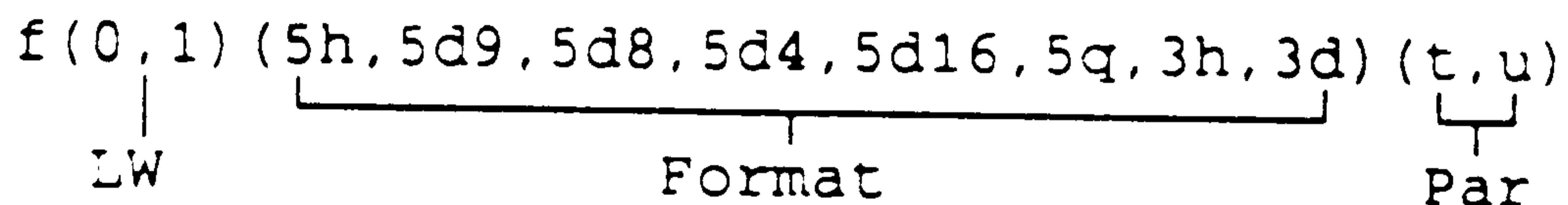
fd* für die Floppy Disk 5 1/4'' und 3 1/2''

Aufbau : (r)fd0135ds18



Weitere Einträge für die Floppy Disks liegen unter /dev/dsk bzw. /dev/rdisk für die Raw-Devices.

Aufbau :



LW - Laufwerk

- Format - 5h = q15d 5.25'' high density disk (1.2MB)
- 5d9 = d9d 5.25'' double density disk (360KB)
 9 Sektoren pro Track
- 5d8 = d8d 5.25'' double density disk (320KB)
 8 Sektoren pro Track
- 5d4 5.25'' double density disk (320KB)
 4 Sektoren pro Track
- 5d16 5.25'' double density disk (320KB)
 16 Sektoren pro Track
- 5q 5.25'' quad density disk (720KB)
- 3h 3.5'' high density disk (1.44MB)
- 3d 3.5'' double density disk (720KB)

- Par - Dieser Parameter selektiert den Diskettenzugriff
 - ohne - gesamte Diskette ausgenommen Zylinder 0
 - t - gesamte Diskette *faste diskette*
 - u - gesamte Diskette ohne Track 0 vom Zylinder 0 *ohne prozess 0 oder andere 0*

USA CE PARI

CP U / DEV / zfd Ø U •
 CP U / zfd Ø / u / dev

COPIA IL CONTENUTO DI FCOPY

DISKI

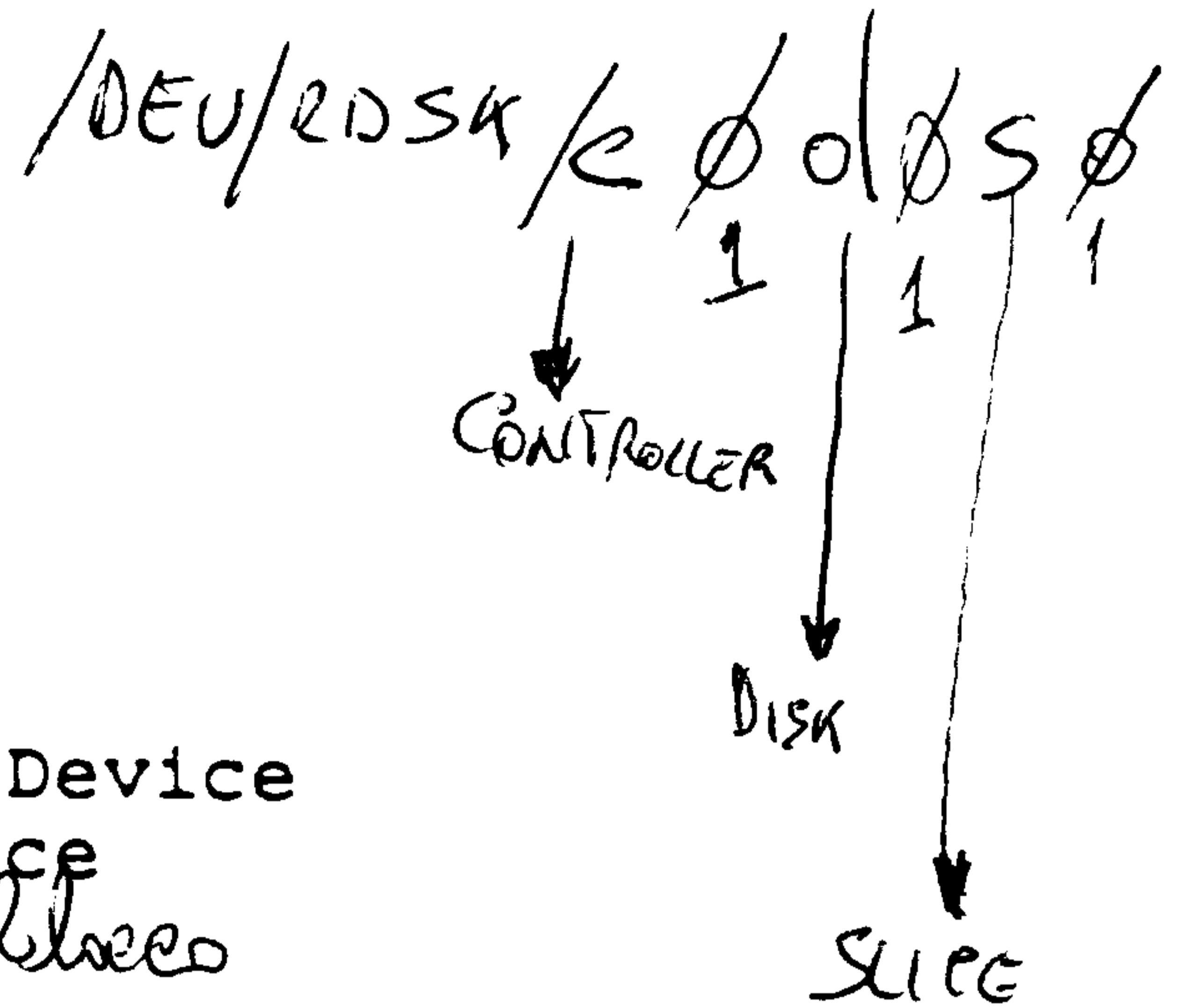
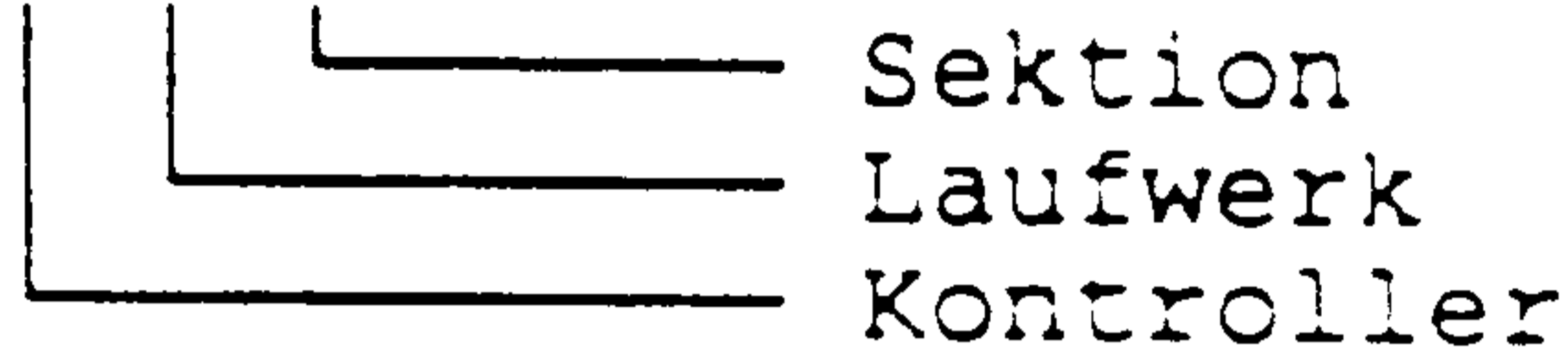
- hd00 -> dsk/c0d0s0 -> dsk/0s0
- hd01 -> dsk/c0d0s3 -> dsk/0s3
- hd02 -> dsk/c0d0s4 -> dsk/0s4
- hd03 -> dsk/c0d0s5 -> dsk/0s5
- hd04 -> dsk/c0d0s6 -> dsk/0s6

SOTTO

Aufbau : /dev/dsk bzw /dec/rdisk

Character Device
 Block Device
 Controller e disco

c?d?s?
 □ □ □

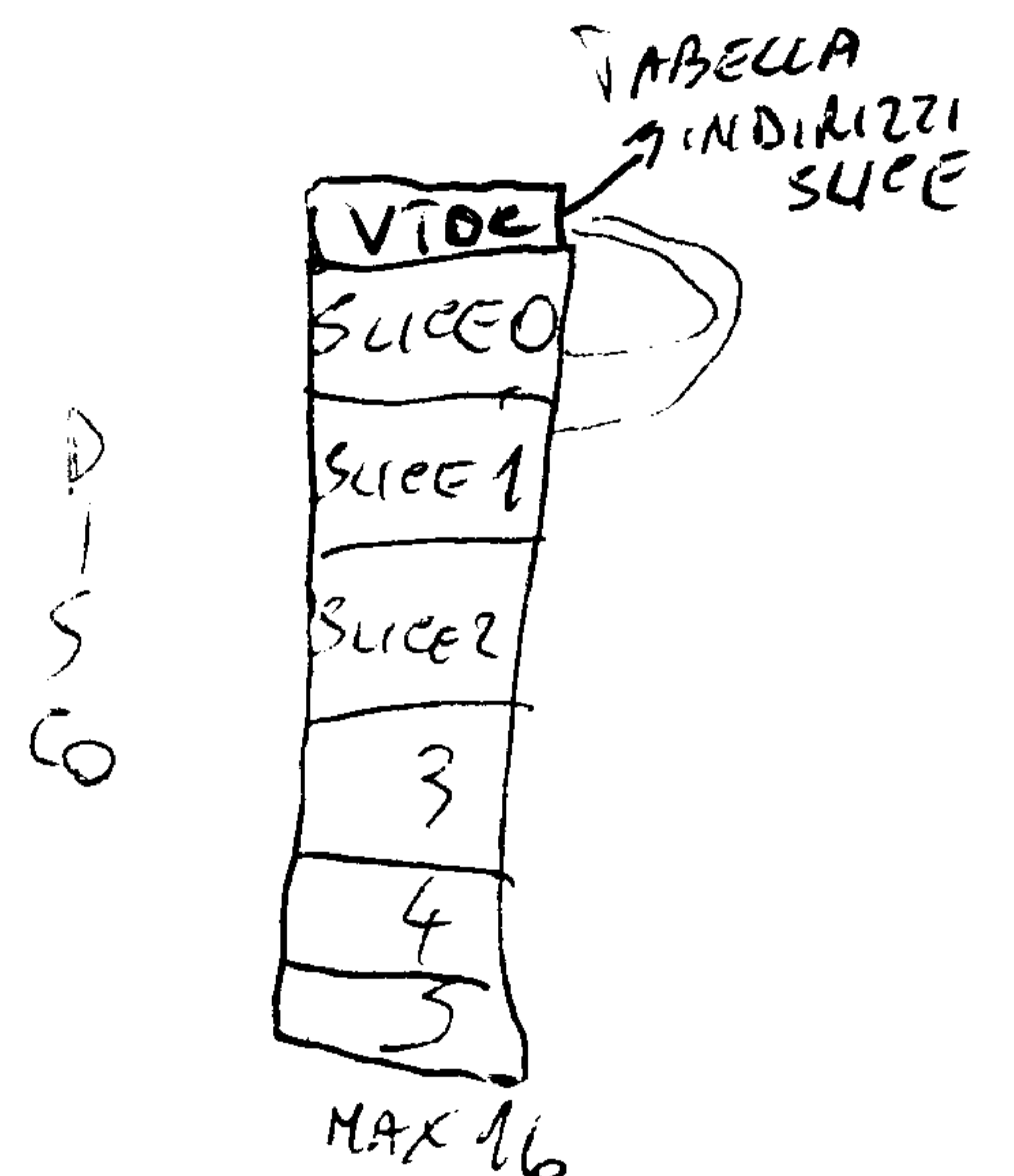


pts??? für die Pseudoterminals

└───┘ Terminalnummer von 000 - 255

tty??? für Drucker- und Terminalanschlüsse

└───┘ Terminalnummer von 000 - 255



4.3 "/etc"-Dateiverzeichnis:

Dieses Dateiverzeichnis enthält die maschinenspezifischen Dateien. Die ausführbaren System-Dateien liegen mit wenigen Ausnahmen (z.B. boot) unter /sbin.

4.4 "/export"-Dateiverzeichnis:

Dieses Datei verzeichnis ist leer.

4.5 "/home" und "/home*"-Dateisysteme

In diesen Dateisystemen werden die Benutzerdaten abgelegt.

SLICE = PARTIZIONI

4.6 "/install"-Dateiverzeichnis

Dieses Dateiverzeichnis ist leer. Es wird bei der Installation von Softwareprodukten benutzt.

4.7 "/lost+found"-Dateiverzeichnis

Ein solches Dateiverzeichnis wird in jedem Dateisystem angelegt, da der fsck dieses zum Ablegen von Dateien nutzt, um Inkonsistenzen im System zu bereinigen.

4.8 "/mnt"-Dateiverzeichnis

Dieses Dateiverzeichnis ist leer. Es wird beim temporären Mounten von Dateisystemen benutzt.

SUBDIVISIONE

4.9 "/opt"-Dateisystem

Das Dateisystem /opt wird zum Ablegen von Anwenderprogrammen benutzt.

Beispiel:

/opt/

	<i>SOFTDIR.</i>	
├──	bin	Siemens Standard Kommandos z.B. HIT, maxed, ft etc.
├──	emds	Emulation von Terminals und Druckern.
├──	etc	Backends für das Siemens Spool System.
├──	ft	Daten und Dateien des Filetransfers.
├──	include	Includedateien der Anwender- Programme, z.B. formant.h
├──	lib	Bibliothekdateien für die Anwender-Programme.
├──	lost+found	Sicherungsverzeichnis für das Kommando fsck.
└──	man	Dateiverzeichnis für Manual- seiten

4.10 "/proc"-Dateisystem

Hier werden alle Dateien zur Prozeßüberwachung vom System hinterlegt.

4.11 "/sbin"-Dateiverzeichnis

Im Dateiverzeichnis /sbin sind alle zentralen Programme für Administration und Operating zu finden

Zu diesen Programmen gehören zum Beispiel :

autopush, bcheckrc, df, disklabel, fsck, init, mount, rc0,
rc1, rc2, rc3, rc6, shutdown, etc.

*level 2 = normale (full) user level 6 = ^{schließen} reboot level 0 = shutdown
// 1-5 = single user level 3 = o/s - n/s (1+2)*

4.12 "/shlib"-Dateiverzeichnis

Dieses Dateiverzeichnis ist leer.

4.13 "/stand"-Dateisystem

Im Dateisystem /stand werden die Default-Werte für Boot hinterlegt. Es gibt folgende Dateien :

/stand/att leere Datei
/stand/boot Default-Werte für den Systemstart
/stand/unix Unix-Betriebssystem

4.14 "/tmp"-Dateisystem

Unter /tmp werden alle temporären Dateien vom System abgelegt.

4.15 "/usr"-Dateisystem

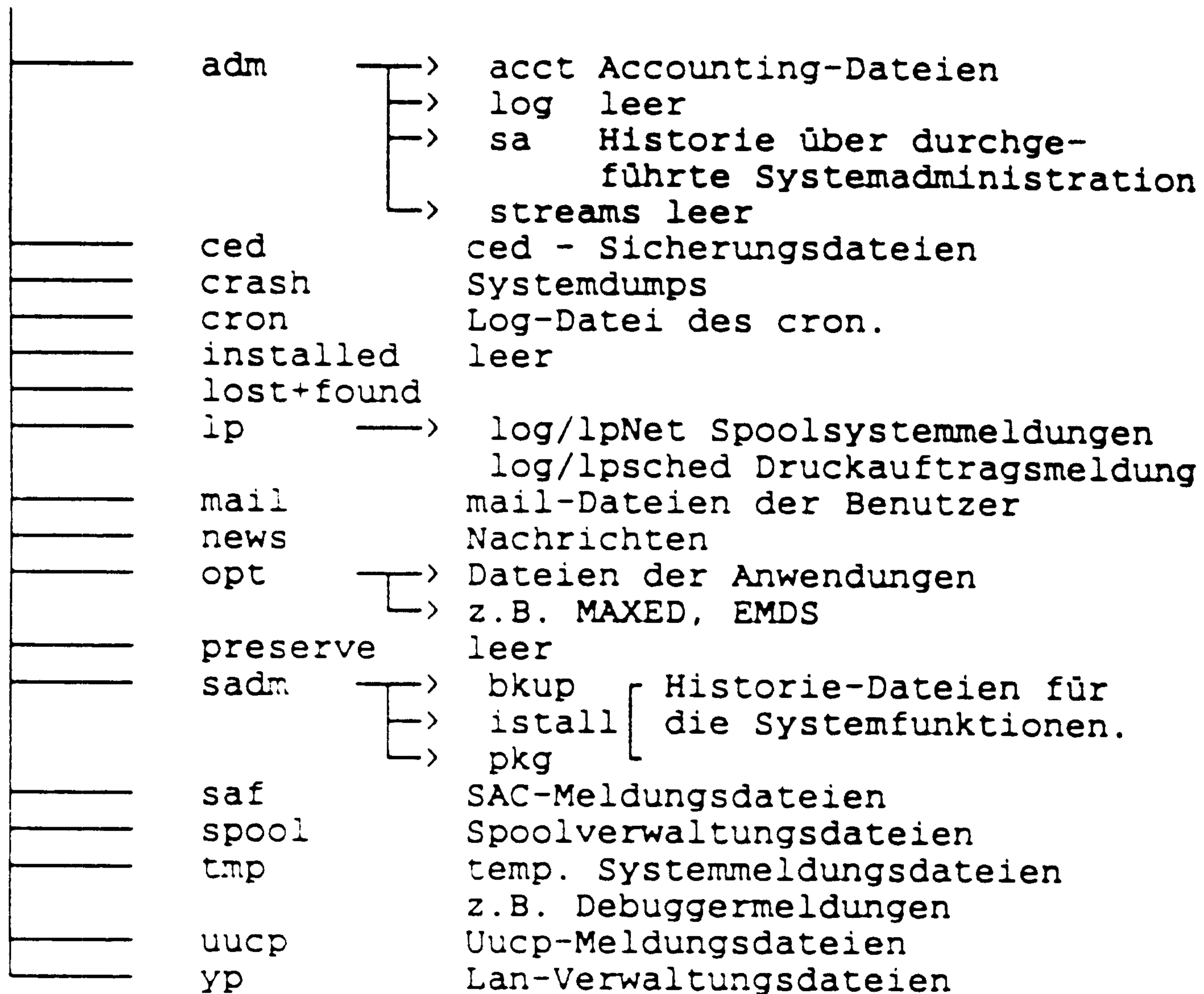
Das Dateisystem /usr ist das Standard-Dateisystem für Systemdateien und Programme.

/usr/		
add-on	—>	/include Include-Dateien
adm		gelinkt auf /var/adm
admin		Systemdateien für Transport- verbindungen
bin		Systemkommandos
ccs	└─>	bin C-Entwicklungssystem (att)
	└─>	lib "
ces	└─>	bin C-Entwicklungssystem (sie)
	└─>	lib "
etc		Umgebungsbeschr. für Monitore
include		Include-Dateien
install		gelinkt auf /home/install
lbin		Programme für Installation
lib		Bibliotheken für Programme und Umgebungsbeschreibungen.
lost+found		
mail		gelinkt auf /var/mail
menus		Menue-Programme für CCP, etc.
net	└─>	servers - Setup für rfs
	└─>	nls - Server und Monitorbeschr.
news		gelinkt auf /var/news
nserve		gelinkt auf /etc/rfs
options		alle Dateien bis auf /rfs.name sind auf /var/sadm/pkg/* gelinkt
preserve		gelinkt auf /var/preserve
pub		gelinkt auf /usr/share/lib
rtmp		
sadm	└─>	bin
	└─>	bkup —> /bin Kommandos
	└─>	install —> /bin Kommandos
	└─>	scripts C-Programm
	└─>	sysadm Systemadmin Menue
sbin		Systemkommandos
share	└─>	man Online-Manuale
	└─>	lib Bibliothek für Programm- und Geräteeigenschaften.
spool		gelinkt auf /var/spool
src	—>	/motif/demos Demo-Sourcen für Motifanwendungen.
tmp		gelinkt auf /var/tmp
ucb		Eine Auswahl der ucb-Kommandos aus der Sinix V5.2x
ucbinclude		ucb-Includedateien
ucb lib		ucb-Bibliotheken
vmsys		Texte, Programme, Konfigurations- dateien und Menus für sysadm

4.16 "/var"-Dateisystem

Das Dateisystem /var ist das Standard-Dateisystem für sich ändernde Dateien und herstellerepezifische Programme.

/var/



Die Kommandos, die von cron auszuführen sind stehen unter /var/spool/cron/crontabs in den Dateien adm root und sys.

Kernelmeldungen werden unter /var/sadm/klog.msg abgelegt.

1

2

3

4

2. Systemdateien

2.1 /etc/vfstab

Die "vfstab" besteht aus 7 verschiedenen Feldern :

special fsckdev mountp fstype fsckpass automnt mntopts

special : Block-Device für lokale Dateisysteme
oder Ressourcenname für Remote Dateisysteme.

fsckdev : Character-Device (korrespondierend zu
special). Wird für den "fsck" benötigt.

mountp : Dateiverzeichnis in das das Dateisystem einge-
hängt wird.

fstype : Typ des Filesystems

fsckpass : Zugangsnummer für die Kommandos ff, fsck, und
ncheck :
"- " verbietet den automatischen Dateisystemcheck
"1" Blöcke beanspruchen mehr als einen i-node.
"2" Blöcke beanspruchen einen i-node außerhalb
des Dateisystems.
"3" Falsche Kettungszähler.
"4" Seiten check.
"5" Falsches i-node-Format.
"6" Blöcke die nicht zugeordnet werden können.
"7" Directory check.
"8" Superblockcheck.

automnt : Mit dem Kommando "mountall" werden alle DVZ's
mit dem Eintrag "yes" gemounted.

mntopts : Liste von Optionen die mit Komma getrennt
werden. Diese Optionen werden beim "mount" ge-
nutzt und beschrieben.

Beispiel
Beispiel :

```
/dev/root      /dev/rroot    /          s5   1   yes  -  
/dev/dsk/0s3   /dev/rdisk/0s3 /usr       s5   1   yes  -  
/dev/dsk/0s4   /dev/rdisk/0s4 /home      ufs  1   yes  -  
/dev/dsk/0s10  /dev/rdisk/0s10 /stand     bfs  1   yes  -  
/dev/dsk/0s11  /dev/rdisk/0s11 /var       ufs  1   yes  -  
/dev/dsk/0s13  /dev/rdisk/0s13 /tmp       ufs  1   yes  -  
/proc -        /proc  proc -    no  -  
/dev/fd -      /dev/fd fdfs -    no  -  
# Diese Einträge werden bei dem Kommando pkgadd genutzt  
/dev/dsk/f0t   /dev/rdisk/f0t /install   s5  -   no  -  
/dev/dsk/f1t   /dev/rdisk/f1t /install   s5  -   no  -  
/dev/dsk/f0     /dev/rdisk/f0  /install   s5  -   no  -  
/dev/dsk/f1     /dev/rdisk/f1  /install   s5  -   no  -
```


2.2 /etc/mnttab

In der Datei "mnttab" stehen alle Dateisysteme, die aktuell eingehängt sind. Die Einträge haben fast das gleiche Format wie in der Datei "vfstab". Nur die Einträge für Rawdevice und die Options für fsckpass fehlen.

Beispiel :

Feld1	Feld2	Feld3	Feld4	Feld5	
					Mountzeitpunkt gerechnet von 1.1.70 in Sekunden
					Mount-Options
					Filesystem-Type
					Mount-DVZ
					Spezial Block-Device (siehe vfstab)

```

/dev/root      /          s5          rw,suid 651317382
/dev/proc      /proc      proc        rw        651317383
/dev/fd        /dev/fd    fdfs        rw        651317383
/dev/dsk/0s11  /var       ufs         suid,rw,noquota 651317385
/dev/dsk/0s3   /usr       s5          rw,suid 651317395
/dev/dsk/0s4   /home      ufs         suid,rw,noquota 651317396
/dev/dsk/0s10  /stand    bfs         rw        651317397
/dev/dsk/0s13  /tmp       ufs         suid,rw,noquota 651317398
  
```

2.3 /etc/termtab

problem der indifferenz der Terminal

In der Datei termtab sind die Initialisierungssequenzen für die Datensichtstationen beschrieben, die konfiguriert sind.

Beispiel:

tty000:is=\E[21u\08 \08\E(K\E[7u:LK=deut

ASPIERATE DES EA

Kennung für die Tastaturtabelle
 Kommando : /etc/keyload *-T System sec*

Steuerfolge, die den Bildschirm und den ladbaren Zeichengenerator initialisiert.

Terminalname

Weitere Informationen im Schnittstellenhandbuch

2.4 /etc/ttytype

x dort ist schon die richtige Form

Die Datei /etc/ttytype wird vom "getty" genutzt.

Beispiel :

97801	pts*	Pseudotys
97801	tty000	

└──┬──┘ Terminalname
└──┬──┘ TERM - Variable

2.5 /etc/termcap

~~example~~

In der Datei "termcap" sind die Eigenschaften von Terminals beim Aufruf bestimmter Anwendungen beschrieben. Die "termcap" wird nur noch aus Kompatibilitätsgründen mitgeliefert. So ist es dem Benutzer möglich, Programme zu nutzen, die auf termcap-Einträge zugreifen. Neue Programme nutzen die Dateien /usr/share/lib/terminfo/?/* (z.B. auch der CED nutzt die "terminfo"-Einträge)

2.6 /usr/share/lib/terminfo /usr/share/lib/ti

Im Datei-Verzeichnis /usr/share/lib/terminfo/? sind die Dateien mit den Eigenschaften von Terminals beim Aufruf von Anwendungen in binärer Form abgelegt.

? bedeutet den ersten Buchstaben des Dateinamens.

Beispiel : vt220

Die Datei ist hinterlegt unter :

/usr/share/lib/terminfo/v/vt220

Die Binärdateien kann man über das Kommando /etc/infocmp disassemblieren.

Beispiel : /etc/infocmp Binärdatei > Sourcedatei

/etc/infocmp vt220 > vt220.ti

- a = Nur in Zusammenhang mit -s. Gibt für alle Benutzer die spezifischen Einträge aus.
- x = Maximalzeit einstellen
- n = Minimalzeit einstellen
- w = Warnzeit einstellen
- l = Benutzerpasswort sperren

user : keine = eigene Benutzerkennung
* = Angabe einer Benutzerkennung

Minimumzeit = der früheste Zeitpunkt in Tage an dem eine Passwortänderung möglich ist.

Maximumzeit = der späteste Zeitpunkt in Tagen an dem eine Passwortänderung erfolgt sein muß. Erfolgt die Änderung nicht wird der Benutzer gesperrt

Warnzeit = Anzahl der Tage vor Maximumzeit an dem der Anwender gewarnt wird, daß er sein Passwort ändern muß.

Beispiel : User-Id "angie"

Kommando : passwd -s angie
Ausgabe : angie PS 10/01/90 1 168 8

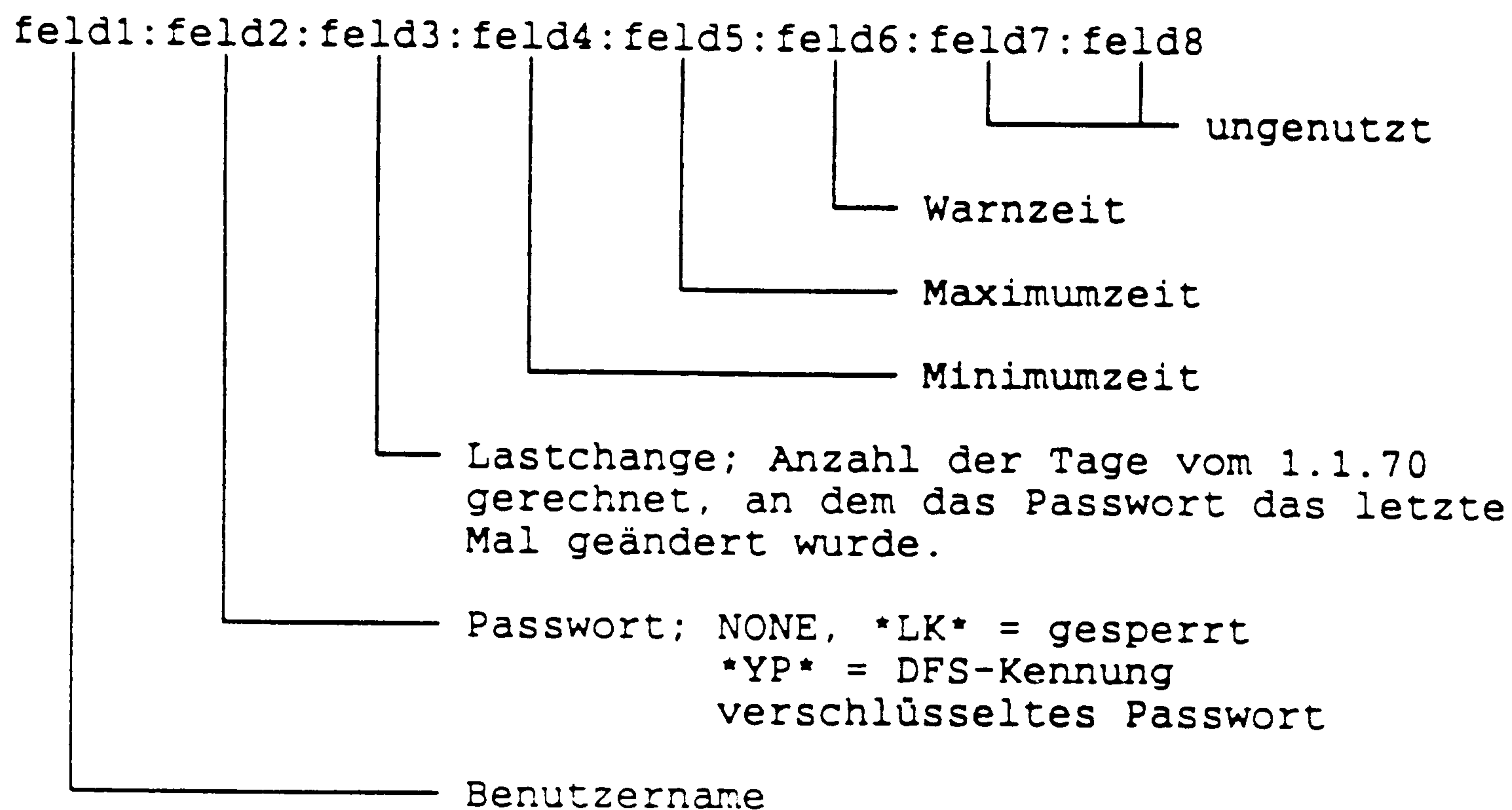
In der Datei "opasswd" wird der vorhergehende Stand gesichert.

2.8 /etc/shadow und oshadow

Beispiel : Datei /etc/shadow

```
root:Ho/hFPMa8CTn6:7518:0:168:7:::  
daemon:NONE:7518:::::  
bin:NONE:7518:::::  
sys:NONE:7518:::::  
adm:NONE:7518:::::  
uucp:NONE:7518:::::  
lp:x:7518:::::  
nuucp::7518:::::
```


Aufbau :



In der Datei "oshadow" wird der vorhergehende Stand gesichert.

2.9 /etc/profile und \$HOME/.profile

Die Datei /etc/profile wird bei einem Sitzungsstart immer durchlaufen. Hier werden unter anderem die Tagesmeldungen (Datei /etc/motd) und die Auslastung der Dateisysteme (Kommando dfspace) ausgegeben.

Benutzereigene Kommandos können in der Datei \$HOME/.profile hinterlegt werden.

2.10 /etc/partitions

In der Datei /etc/partitions werden die Werte der Platteneinteilung hinterlegt. Diese werden bei der Installation vom Systemverwalter festgelegt.

Beispiel :

disk00:
heads=15, cyls=1224, sectors=34, bpsec=512, vtocsec=0,
altsec=1, boot="/etc/boot", device="/dev/rdisk/c0d0s0

root:
partition = 1, start = 77554, size 63750,
tag = ROOT, perm = VALID

swap:
partition = 2, start = 11254, size 66300,
tag = SWAP, perm = NOMOUNT , perm = VALID

usr :
partition = 3, start = 141304, size = 142800
tag = USR, perm = VALID

reserved: partition = 7,

2.11 /etc/ap/chan.ap

Kommando : /sbin/autopush

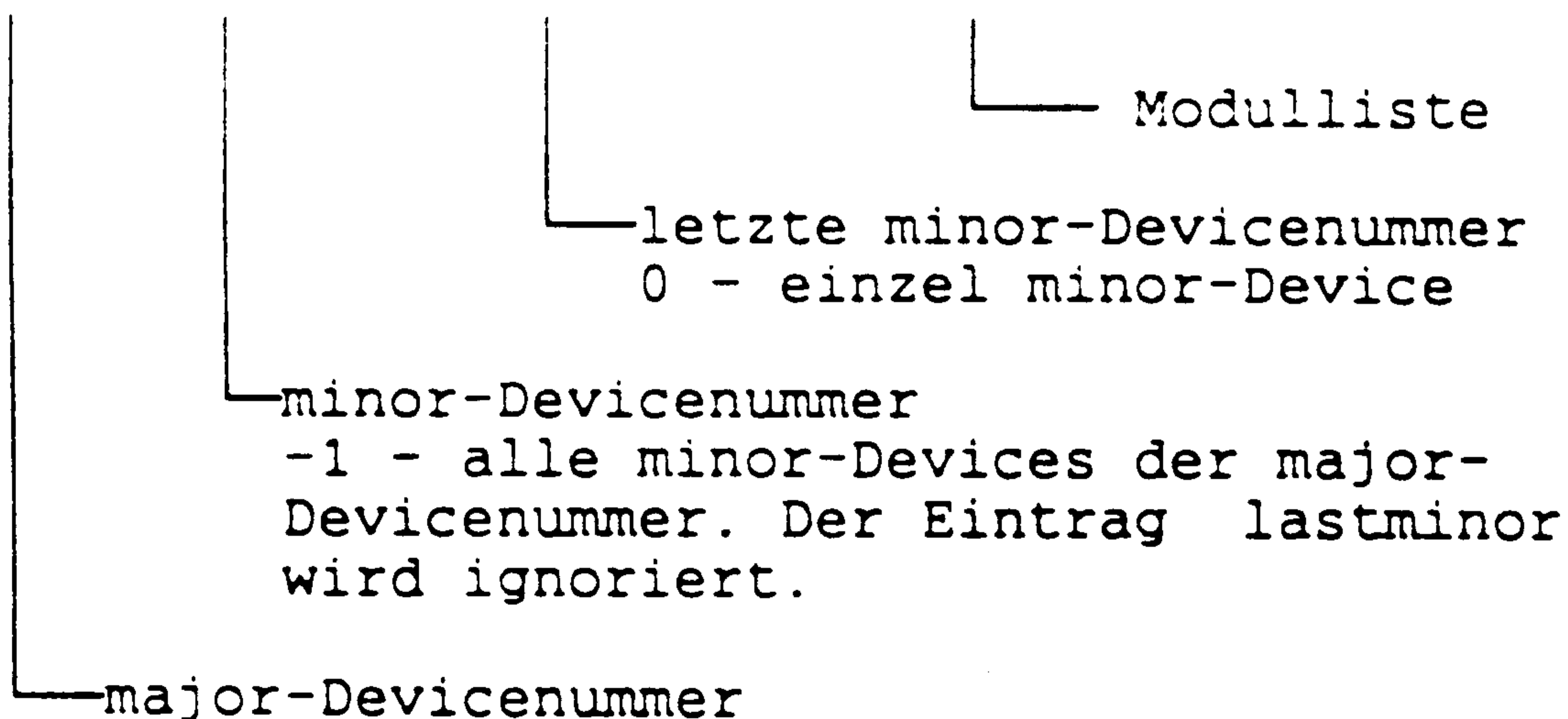
Kapitel 1.4.2

Beispiel :

```
#ident "$(#)autopush:i386/chan.ap 1.7"  
#ident "$Header: chan.ap 2.1 90/05/18 $"
```

```
# /dev/console and /dev/contry autopush setup
```

```
#  
# major minor lastminor modules  
5 -1 0 char ansi ldterm ttcompat  
3 0 129 ldterm ttcompat
```



Mann kann diese Technik nutzen um Codeumsetzung oder
andere einem Device vorgeschaltete Aufgaben durchzuführen
z.B. ldterm = realisiert die Linedisziplin

1

2

3

4

COMANDOS

Kommandos in SINIX V5.4

VI

||

ced Datei

Edieren von Datei mit max. 512 Zeichen pro Zeile

ztl - x b

hd Datei

Datei wird in Hex/Ascii angezeigt (früher xd)

x liegen in erodechnele

strings Datei

nur darstellbare Zeichen werden angezeigt

simulna i simulnschli

Welle

who -r

Welcher Runlevel ?

whodo

Wer macht was ?

finger

Information über alle Benutzer

passwd

Passwort : min. 6 Zeichen
min. 2 Buchstaben
min. 1 Nummer oder 1 Sonderz.

sh

Bourne-Shell

ksh *= SH*

Korn-Shell

jsh

Job-Control-Shell

csh

C-Shell

priontl

Zeitscheibenzuordnung (realtime/normal)

sar -o Datei t n

Systemaktivitäten werden in Datei geschrieben :
n-Messungen im Zeitintervall von je t-Sekunden

sar -f Datei

alle Informationen über die Systemaktivitäten aus der
Datei (sar-Format) werden angezeigt

Texteditor vi

1. Kurzbeschreibung

Der vi ist ein bildschirmorientierter Editor (vi=visual). Er stellt eine erweiterte Form des Zeileneditors ex dar.

- Besonderheiten :
- standard in Unix-Systemen
 - hoher Lernaufwand
 - mächtige Befehlssprache : reguläre Ausdrücke suchen und ersetzen
 - drei verschiedene Modis : Kommando-Modus
Eingabe-Modus
Statuszeilen-Modus

2. Übersicht der wichtigsten Kommandos

a) Start : vi Datei

b) Bedienung :

Kommando-Modus			Eingabe-Modus
a	append	----->	Eingabe-Modus Text
		<-----	ESC-Taste
i	insert	----->	Eingabe-Modus Text
		<-----	ESC-Taste
dd	Zeile loeschen		
x	Zeichen loeschen		
^f	naechste Seite		
^b	vorherige Sete		
nG	Position n-te Zeile		
:	ex-Kommandos	----->	Statuszeilen-Modus Statuszeile
			! Kommando Systemkommando
			set nu Zeilennummern
			1,\$s/alt/neu/g Ersetzen
			q! nur exit
			wq sichern+exit
		<-----	CR-Taste
/	Suchkommandos	----->	Statuszeile
			Muster Muster eingeben
		<-----	CR-Taste
ZZ	sichern+exit		

Systemaktivitäten : sar

Das Kommando sar:

Die Kommandos sar und sag geben Informationen über die Systemaktivitäten aus. Die Tagesdatendatei steht im Dateiverzeichnis /usr/adm/sa. In den Tagesdatendateien stehen nur wenige Informationen. Daher ist es sinnvoll mit dem Kommando sar zunächst eine Datei zu erzeugen, die dann mit weiteren sar-Kommandos ausgewertet wird.

Die Datei wird folgendermaßen erzeugt:

Beispiel: sar -o ausgabe 60 10 *(Messung)*
Anzahl der Messungen (10) für
einen Zeitraum (jeweils 60 s)
Zeitraum für den eine Messung
erfolgt (60 s)
Name der Ausgabedatei

sar -b -f ausgabe *(Ausgabe)*

-b Ausgaben zur Systempufferauslastung

bread/s	Datenübertragung zwischen Systempuffer und Festplatte; Lesezugriff auf Festplatte
bwrit/s	Schreibzugriff auf Festplatte
lread/s	Lesezugriffe auf den Systempuffer
lwrit/s	Schreibzugriffe auf den Systempuffer
%rcache	Lesezugriffe, die aus dem Puffer erledigt werden können.
%wcache	Schreibzugriffe, die aus dem Puffer erledigt werden können.
pread/s	Datenübertragungen über raw-Devices; Lesezugriffe
pwrit/s	Schreibzugriffe

-c Systemaufrufe

scall/s	Anzahl der Systemaufrufe pro Sekunde
sread, swrit, fork, exec	Anzahl der Systemaufrufe read, write, fork, exec pro Sekunde
rchar/s	Anzahl der Zeichen, die durch read-Systemaufrufe übertragen werden
wchar/s	Anzahl der Zeichen, die durch write-Systemaufrufe übertragen werden.

Systemaktivitäten : sar

-g Paging-Aktivitäten

pgout/s Anzahl der Anforderungen zum Auslagern von Pages pro Sekunde

ppgout/s Anzahl der Pages, die pro Sekunde tatsächlich ausgelagert wurden

pgfree/s Anzahl der Pages, die pro Sekunde frei werden

-p Paging

pgin/s Anzahl der Anforderungen pro Sekunde zum Einlagern von Seiten in den Hauptspeicher

ppgin/s Anzahl der Pages, die pro Sekunde eingelagert wurden.

vflt/s Anzahl der Zugriffe pro Sekunde, bei der die gültige Seite nicht im Hauptspeicher gefunden wurde

pflt/s, slock/s weitere Zugriffsfehler

-r nicht benutzte Speicherplatzseiten und Festplattenblöcke

freemen durchschnittliche Anzahl von Pages, die für Benutzer-Prozesse zur Verfügung stehen

freeswap Anzahl der Festplattenblöcke (512 Byte), die für die Auslagerung von Seiten zur Verfügung stehen

-k Speicher-Reservierung durch den Betriebssystemkerns
Es wird Speicherplatz für kleine, große und sehr große Speicherplatzanforderungen zur Verfügung gestellt.

sml_mem Speicherplatz in Byte, der für kleine Speicherplatzanforderungen zur Verfügung steht

alloc Speicherplatz in Byte, der durch kleine Speicherplatzanforderungen belegt ist.

fail Anzahl der kleinen Speicherplatzanforderungen, die nicht erledigt werden konnten.

lg_mem Speicherplatz in Byte, der für große Speicherplatzanforderungen zur Verfügung steht.

alloc

fail

Systemaktivitäten : sar

ovsz_alloc Speicherplatz, der durch sehr große Speicherplatzanforderungen belegt ist.

fail

-q durchschnittliche Warteschlangenlängen

runq-sz Anzahl der Prozesse, die ablauffähig im Hauptspeicher stehen

-v Systemtabellen

proc-sz Anzahl der tatsächlichen/möglichen Einträge in der Prozeßtabelle

inod-sz Anzahl der tatsächlichen/möglichen Einträge in der Inode-Tabelle

file-sz unvollständige Ausgabe!
Anzahl der offenen Dateien

lock-sz Anzahl der tatsächlichen/möglichen Einträge in der Locking-Tabelle

Swapping: sar -w *U 3 10*

nelke20 nelke20 3.2 2 1386 06/23/90

09:40:29	swpin/s	bswin/s	swpot/s	bswot/s	pswch/s
12:00:01	0,00	0,0	0,00	0,0	1
12:00:01	0,00	0,0	0,00	0,0	60
12:20:01	0,00	0,0	0,00	0,0	2
12:20:01	0,00	0,0	0,00	0,0	36
12:40:01	0,00	0,0	0,00	0,0	1
12:40:01	0,00	0,0	0,00	0,0	64
13:00:02	0,00	0,0	0,00	0,0	2
13:00:02	0,00	0,0	0,00	0,0	67
13:20:02	0,00	0,0	0,01	0,1	4
13:20:02	0,00	0,0	0,00	0,0	75
Average	0,00	0,0	0,00	0,0	2

swpin/s Anzahl der Transfers in den Speicher je Sekunde

bswin/s Anzahl der Blöcke für swpin je Sekunde

swpot/s Auslagerungen in den Swap-Bereich. Ist der Wert größer als 1 Sekunde, muß mehr Speicher her.

bswot/s Anzahl der gewappten Blöcke je Sekunde

pswch/s Anzahl der Prozeßwechsel je Sekunde

Kommando-Übersicht

KOMMANDO-INTERPRETER

csch	C-Shell
jsh	Bourne-Shell mit Auftragssteuerung (job control)
ksh	Korn-Shell
rksh	Eingeschränkte Korn-Shell
rsh	Shell-Kommando am fernen Rechner ausführen
sh	Bourne-Shell

EINGEBAUTE KOMMANDOS DER BOURNE-SHELL SH

Wenn nicht anders angegeben, finden Sie die eingebauten Kommandos unter ihrem Namen beschrieben.

break	Schleife abbrechen (siehe <i>sh</i>)
cd	Aktuelles Dateiverzeichnis wechseln
continue	Schleife abbrechen (siehe <i>sh</i>)
echo	Aufruf-Argumente ausgeben
eval	Aufruf-Argumente bearbeiten und als Kommando ausführen
exec	Die aktuelle Shell überlagern
exit	Shell-Prozedur beenden
export	Shell-Variablen exportieren
getopts	Argumente einer Prozedur nach Optionen durchsuchen
hash	Hash-Tabelle der Shell bearbeiten
newgrp	Gruppenzugehörigkeit ändern
pwd	Pfadnamen des aktuellen Dateiverzeichnisses ausgeben
read	Argumente von der Standard-Eingabe lesen und Shell-Variablen zuweisen
readonly	Shell-Variablen schützen
return	Shell-Funktion beenden (siehe <i>sh</i>)
set	Shell-Optionen oder Stellungsparameter setzen
shift	Die Werte der Stellungsparameter nach links verschieben
test	Bedingungen prüfen
times	Gesamt-Laufzeit der bisher gestarteten Prozesse ausgeben
trap	Signalbehandlung ändern
type	Typ eines Kommandos abfragen
ulimit	Datei-Größe für das Schreiben begrenzen oder aktuellen Grenzwert abfragen
umask	Standardvergabe der Zugriffsrechte ändern
unset	Shell-Variablen oder Shell-Funktionen aus der Umgebung löschen
wait	Auf die Beendigung von Hintergrund-Prozessen warten
.	Shell-Prozeduren in der aktuellen Shell ausführen
:	Ende-Status 0 zurückgeben
[.....]	Bedingungen prüfen (wie <i>test</i>)

BENUTZERUMGEBUNG ABFRAGEN BZW. ÄNDERN

cd	Aktuelles Dateiverzeichnis wechseln
env	Umgebung bei Ausführung von Kommandos ändern
finger	Informationen über Benutzer am lokalen und fernen System ausgeben
hostname	Rechnernamen ausgeben und festlegen
id	Benutzer- und Gruppennummer und zugehörige Kennung ausgeben
logname	Login-Kennung abfragen
ls	Informationen über Dateiverzeichnisse und Dateien ausgeben
pwd	Pfadnamen des aktuellen Dateiverzeichnisses ausgeben
tty	Pfadnamen der aktuellen Datensichtstation ausgeben
who	Aktive Benutzerkennungen anzeigen

Kommando-Übersicht

bearbeiten

awk	Programmierbare Bearbeitung von Textdateien
bdiff	Große Dateien vergleichen
cmp	Dateien zeichenweise vergleichen
comm	Gleiche Zeilen in zwei sortierten Dateien suchen
crypt	Text verschlüsseln und entschlüsseln
csplit	Datei nach bestimmten Kriterien unterteilen
cut	Felder oder Spalten aus den Zeilen einer Datei ausschneiden
deroff	Herausfiltern von nroff-, troff-, tbl- und eqn-Anweisungen
diff	Dateien zeilenweise vergleichen
diff3	Drei Dateien zeilenweise vergleichen
dircmp	Dateiverzeichnisse vergleichen
egrep	Muster suchen
fgrep	Zeichenketten suchen
find	Dateiverzeichnisse durchsuchen
fmt	Einfache Textformatierung
fold	Lange Zeilen zerlegen
grep	Muster suchen
hashcheck	Rechtschreibliste überprüfen
hashmake	Hash-Liste erstellen
join	Zwei Dateien nach Vergleichsfeldern verbinden
makekey	Code für Verschlüsselung festlegen
nl	Textzeilen numerieren
newform	Format einer Textdatei ändern
paste	Zeilen zusammenfügen
sdiff	Dateien vergleichen und nebeneinander ausgeben
sort	Dateien sortieren und/oder mischen
spell	Rechtschreibfehler suchen
spellin	Komprimierte Wortliste erzeugen
split	Datei auf mehrere Dateien verteilen
sum	Prüfsumme einer Datei berechnen
tr	Zeichen ersetzen oder löschen
uniq	Mehrfache Zeilen suchen
wc	Wörter, Zeichen und Zeilen zählen

sichern und archivieren

ar	Bibliotheken verwalten
cpio	Dateien und Dateiverzeichnisse ein- und auslagern
dd	Dateien kopieren und konvertieren
tar	Archivieren von Dateien auf Magnetband-Kassette, Band oder Diskette und Archive bearbeiten

komprimiert speichern bzw. in Ursprungszustand zurückversetzen

compress	Dateien komprimieren
pack	Dateien komprimieren
pcat	Komprimierte Dateien ausgeben
uncompress	Komprimierte Dateien expandieren
unpack	Komprimierte Dateien expandieren
zcat	Komprimierte Dateien ausgeben

Kommando-Übersicht

Dateieigenschaften abfragen und ändern

chgrp	Gruppennummer einer Datei ändern
chmod	Zugriffsrechte ändern
chown	Eigentümer einer Datei ändern
file	Art einer Datei bestimmen
ls	Informationen über Dateiverzeichnisse und Dateien ausgeben
touch	Änderungs- und Zugriffszeiten aktualisieren
umask	Standardvergabe der Zugriffsrechte ändern

Dateisystem verändern

copy	Dateien gruppenweise kopieren
cp	Dateien kopieren
cpio	Dateien und Dateiverzeichnisse ein- und auslagern
destroy	Dateien physikalisch löschen
find	Dateiverzeichnisse durchsuchen
ln	Verweis auf eine Datei eintragen
mkdir	Dateiverzeichnis erzeugen
mv	Dateien versetzen oder umbenennen
rm	Dateien löschen
rmdir	Dateiverzeichnisse löschen
split	Datei auf mehrere Dateien verteilen

EDITOREN

bfs	Große Dateien durchsuchen
ced	Bildschirmorientierter Editor
ctags	Markierungsdatei erstellen
ed	Zeilenorientierter Editor im Dialogbetrieb
edit	Einfach zu bedienender Editor
ex	Zeilenorientierter Editor
red	Eingeschränkter zeilenorientierter Editor im Dialogbetrieb
sed	Editor im Prozedurbetrieb
vi	Bildschirmorientierter Editor

HILFSKOMMANDOS FÜR SHELL-PROZEDUREN

basename	Dateinamen vom Pfad trennen
dirname	Pfad-Präfix vom Dateinamen trennen
expr	Ausdrücke auswerten
false	Ende-Status ungleich 0 zurückgeben
getopt	Argumente einer Prozedur nach Optionen durchsuchen
getoptcv	getopt-Kommandoaufrufe in getopt-Kommandoaufrufe konvertieren
getopts	Argumente einer Prozedur nach Optionen durchsuchen
sleep	Prozesse zeitweise stilllegen
test	Bedingungen prüfen
true	Ende-Status 0 zurückgeben
xargs	Argumentliste aufbauen und Kommando ausführen
[...]	Bedingungen prüfen (wie <i>test</i>)

Kommando-Übersicht

DRUCKEN UND DRUCKERVERWALTUNG

cancel	Druckaufträge löschen
disable	Drucker deaktivieren
enable	Drucker aktivieren
lp	Dateien ausdrucken
lpr	Dateien ausdrucken und Druckaufträge steuern
lpstat	Informationen über Druckaufträge ausgeben
pr	Dateien für Ausgabe aufbereiten

ZEICHEN EINLESEN UND AUSGEBEN

echo	Aufruf-Argumente ausgeben
line	Eine Zeile lesen
printf	Formatierte Ausgabe
tee	Pipes zusammenfügen und Eingabe kopieren

ZEICHEN EINLESEN, UMWANDELN UND AUSGEBEN

banner	Zeichen in großer Darstellung ausgeben
col	Filter für umgekehrte Zeilenvorschübe
hd	Dateiinhalte hexadezimal ausgeben
od	Dateiinhalte oktal ausgeben

MELDUNGEN FORMATIEREN

fmtmsg	Meldung formatiert ausgeben
--------	-----------------------------

BENUTZEREIGENSCHAFTEN ABFRAGEN UND ÄNDERN

groups	Gruppe eines Benutzers ausgeben
id	Benutzer- und Gruppennummer und zugehörige Kennung ausgeben
login	Sich neu am System anmelden
logname	Login-Kennung abfragen
mesg	Nachrichtenempfang verbieten oder erlauben
newgrp	Gruppenzugehörigkeit ändern
passwd	Login-Kennwort und Kennwortattribute eintragen oder ändern
su	Benutzerkennung vorübergehend wechseln

BENUTZER VERWALTEN

finger	Informationen über Benutzer am lokalen und fernen System ausgeben
groups	Gruppe eines Benutzers ausgeben
last	Letzten Benutzer oder letztes Terminal-Login ausgeben
listusers	Informationen über Benutzerkennungen ausgeben
passwd	Login-Kennwort eintragen oder ändern
rwho	Aktive Benutzerkennungen im Netz anzeigen
who	Aktive Benutzerkennungen anzeigen
whois	Internet-Service zum Auffinden von Benutzerkennungs-Dateiverzeichnissen

Kommando-Übersicht

KOMMUNIKATION MIT ANDEREN BENUTZERN

mail	Nachrichten senden oder lesen
mailalias	Aliasnamen für Mailadressen umsetzen
mailx	Nachrichten interaktiv bearbeiten
mesg	Nachrichtempfang verbieten oder erlauben
news	Nachrichten ausgeben
notify	Ankunft neuer Post melden
rmail	Nachrichten senden
talk	Dialog mit anderem Benutzer führen
vacation	Post automatisch speichern und beantworten
write	Nachricht an einen Benutzer senden

KALENDERFUNKTIONEN UND TERMINE

at	Kommandos zu einem späteren Zeitpunkt ausführen
atq	Kommandoaufträge, die mit at oder batch erteilt wurden, auflisten
atrm	Kommandoaufträge, die mit at oder batch erteilt wurden, löschen
cal	Kalender ausgeben
calendar	Terminkalender
crontab	Kommandos regelmäßig zu bestimmten Zeitpunkten ausführen
date	Datum und Uhrzeit ausgeben oder Systemuhr stellen

RECHENFUNKTIONEN

bc	Arithmetische Sprache
dc	Tischrechner
expr	Ausdrücke auswerten
factor	Zahl in ihre Primfaktoren zerlegen
units	Einheiten umrechnen

INFORMATIONEN ÜBER PROZESSE

acctcom	Prozeßabrechnungsstatistiken suchen und drucken
gcore	Speicherabzug von laufenden Prozessen erstellen
priocntl	Zeitscheibenverteilung und Prioritäten einstellen
ps	Prozeßdaten abfragen
time	Laufzeit eines Kommandos messen
times	Gesamt-Laufzeit der bisher gestarteten Prozesse ausgeben
timex	Laufzeit eines Kommandos messen, Prozeßdaten und Systemaktivitäten anzeigen
truss	Systemaufrufe und Signale protokollieren

PROZESSE STEUERN

batch	Kommandos zu einer späteren Zeit ausführen
kill	Signale an Prozesse senden
nice	Priorität von Kommandos ändern
nohup	Kommando ausführen und dabei Signale ignorieren
priocntl	Zeitscheibenverteilung und Prioritäten einstellen
sleep	Prozesse zeitweise stilllegen
wait	Auf die Beendigung von Hintergrund-Prozessen warten

Kommando-Übersicht

INTERPROZESSKOMMUNIKATION

ipcrm Einrichtungen zur Interprozeß-Kommunikation entfernen
ipcs Zustand von Interprozeß-Kommunikationseinrichtungen
 ausgeben
mknod Benannte Pipe erzeugen

DATENSICHTSTATION

clear Bildschirm löschen
keyload Tastaturtabellen laden
last Letzten Benutzer oder letztes Terminal-Login ausgeben
script Sitzung protokollieren
stty Eigenschaften einer Datensichtstation ausgeben oder
 ändern
tabs Tabulatorstops setzen
tput Datensichtstation initialisieren oder Datenbasis *terminfo*
 abfragen
tty Pfadnamen der aktuellen Datensichtstation ausgeben

SPEICHERPLATZBELEGUNG ÜBERPRÜFEN

df Dateisystem auf freien Platz prüfen
du Belegten Speicherplatz ausgeben

INFORMATIONEN ÜBER SYSTEMDATEN

finger Informationen über Benutzer am lokalen und fernen System
 ausgeben
last Letzten Benutzer oder letztes Terminal-Login ausgeben
ps Prozeßdaten abfragen
sag Systemaktivität graphisch anzeigen
sar über Systemtätigkeit berichten
sysname Informationen zum Betriebssystem ausgeben
timex Laufzeit eines Kommandos messen, Prozeßdaten und
 Systemaktivitäten anzeigen
uname Namen des aktuellen Systems ausgeben
who Aktive Benutzerkennungen anzeigen

INFORMATION ÜBER PROZESSORTYP

i386 Wahrheitswert über Prozessoridentität zurückgeben

ON-LINE-DOKUMENTATION

man On-Line-Dokumentation nutzen

SYSTEMPUFFER LEEREN

sync Systempuffer zurückschreiben

KONFIGURATION EINES DATENSTROMS

strchg Konfiguration eines Datenstroms ändern
strconf Konfiguration eines Datenstroms abfragen

Kommando-Übersicht

DISKETTEN UND MAGNETBÄNDER BEARBEITEN

flchk	Labelbereich einer Diskette überprüfen
fldisp	Labelbereich einer Diskette ausgeben
flinit	In den Labelbereich einer Diskette schreiben
format	Disketten formatieren
mt	Magnetband bearbeiten

FENSTERVERWALTUNG

ismpx	Zustand eines Bildschirms mit Fensterdarstellung abfragen
jterm	Shell-Fenster auf einem Bildschirm mit Fensterdarstellung zurücksetzen
jwin	Größe eines Shell-Fensters abfragen
layers	Multiplexer für Bildschirme mit Fensterdarstellung
relogin	Aktuelles Shell-Fenster als login-Eintrag definieren
shl	Schichtenverwaltung für Shells

FORMULAR- UND MENÜSPRACHEN-INTERPRETER FMLI

fmlü	FMLI aktivieren
------	-----------------

FACE-MENÜ (FRAMED ACCESS COMMAND ENVIRONMENT)

face	FACE-Menü eröffnen
------	--------------------

NETZ-KOMMANDOS

ct	getty-Prozeß für ferne Datensichtstation erzeugen
cu	Verbindung zu einem anderen UNIX-System aufbauen
finger	Informationen über Benutzer am lokalen und fernen System ausgeben
ftp	Programm zur Dateiübertragung
rcp	Datei von oder zu einem fernen Rechner kopieren
rlogin	An einem fernen Rechner anmelden
ruptime	Zustand der Rechner im lokalen Netz anzeigen
rwho	Aktive Benutzerkennungen im Netz anzeigen
telnet	Benutzerschnittstelle zum TELNET-Protokoll
tftp	Einfaches Dateiübertragungs-Programm
uucp	Dateien zwischen Unix-Systemen kopieren
uudecode	Datei nach der Übertragung per <i>mail</i> decodieren
uuencode	Datei für die Übertragung per <i>mail</i> codieren
uuglist	Service-Liste angeschlossener UNIX-Rechner
uulog	UUCP-Protokolldateien ausgeben
uuname	Namen von UUCP-Systemen auflisten
uupick	Dateiübertragung zwischen UNIX-Rechnern
uustat	Kontroll-Funktion zur Dateiübertragung in öffentlichen UNIX-Systemen
uuto	Dateiübertragung zwischen UNIX-Rechnern
uux	Kommando auf fernem System ausführen
whois	Internet-Service zum Auffinden von Benutzerkennungs-Dateiverzeichnissen

Kommando-Übersicht

MS-DOS-DATEIEN UND -DISKETTEN BEARBEITEN

doscat	Dateien einer MS-DOS-Diskette ausgeben
doscpc	Dateien zwischen SINIX und MS-DOS kopieren
dosdir	Inhalt von Dateiverzeichnissen einer MS-DOS-Diskette im MS-DOS-Format ausgeben
dosfilt	Dateien mit sprachabhängigen Sonderzeichen für MS-DOS lesbar machen
dosformat	MS-DOS-Diskette formatieren
dosls	Inhalt von Dateiverzeichnissen einer MS-DOS-Diskette im SINIX-Format ausgeben
dosmkdir	Dateiverzeichnisse auf einer MS-DOS-Diskette einrichten
dosrm	MS-DOS-Dateien löschen
dosrmdir	Dateiverzeichnisse im MS-DOS-Dateisystem löschen
sinfilt	Dateien mit sprachabhängigen Sonderzeichen für SINIX lesbar machen

NLS-KOMMANDOS (NATIVE LANGUAGE SYSTEM)

dumpmsg	Meldungstext-Datei aus Meldungskatalog-Datei erzeugen
extr	Zeichenketten in Quellprogrammen suchen und ersetzen
extract	Zeichenketten in Quellprogrammen interaktiv suchen und ersetzen
gencat	Binär codierten Meldungskatalog erzeugen
gettxt	Zeichenketten in einer Datenbasis für Meldungstexte auffinden
ic	Internationale Datenbasis übersetzen
iconv	Code konvertieren
mkmsgs	Meldungsdateien für gettxt erstellen
srchtxt	Inhalt von Meldungsdateien anzeigen, nach Zeichenketten suchen

I N H A L T S V E R Z E I C H N I S

- 1. Systemstart
 - 1.1 Ablauf eines Systemstarts
 - 1.2 Der Boot
 - 1.3 init
 - 1.3.1 inittab
 - 1.3.1.1 Identifikation
 - 1.3.1.2 Run-Level
 - 1.3.1.3 Schlüsselwort
 - 1.3.1.4 Kommandos
 - 1.4 Beispiel der "inittab"
 - 1.4.1 /sbin/chkconsole
 - 1.4.2 /sbin/autopush
 - 1.4.3 /sbin/setcik
 - 1.4.4 /sbin/bcheckrc
 - 1.4.5 /sbin/rc? - Prozeduren
 - 1.4.6 /sbin/uadmin
 - 1.4.7 /usr/bin/lm
 - 1.4.8 /usr/lib/saf/sac
 - 1.4.9 /etc/getty
- 2. Dateisystemtypen
 - 2.1 Platteneinteilung
 - 2.2 s5 - System
 - 2.3 ufs - System
 - 2.4 bfs - System

1. Systemstart

1.1 Ablauf eines Systemstarts

7bit od
seine parte

boot

Datei : /stand/boot



init

Datei : /etc/inittab



chkconsole

← console (= Modem & console)



autopush

gesteuert Terminal
Datei : /etc/ap/chan.ap



setclk

→ de organo d machine



bcheckrc

→ che sur file sistema
Datei : /etc/vfstab → fstab → 4/2/0
/etc/mnttab



rc

Datei : /sbin/rc0 de inittab
/sbin/rc1
/sbin/rc2
/sbin/rc3
/sbin/rc6



run-level

1.2 Der Boot

Über das Boot-Programm wird das stand-alone Unix interaktiv geladen und gestartet. Es ist jedoch ebenso möglich, andere stand-alone Programme zu laden und zu starten (z.B. TDS). Das Boot-Programm muß ein Teil des UNIX BASE Operating System und im "root"-Dateiverzeichnis vorhanden sein, um ein sicheres Laden des Kerns zu gewährleisten. Während der Installation des Unix Operating System wird ein herstellerepezifischer Masterboot auf dem Sektor 0 auf der Platte hinterlegt. Er ist das Default-Boot-Programm für die Plattenbootprozedur.

Das System startet das ROM-Boot-Programm immer wenn die Maschine eingeschaltet wird. Als erstes wird vom Boot versucht, ob er eine bootfähige Floppy findet. Ist dieses nicht der Fall versucht der Boot über die Platte einen Ladevorgang.

Die Floppy Disk Boot-Prozedur hat zwei Stufen :

1. Über den Boot-Block auf dem Sektor 0 des Dateisystems wird der Boot geladen.
2. Der Boot wird gestartet und wartet auf Eingaben des Benutzers.

Die Platten Boot-Prozedur hat drei Stufen :

1. Über den ROM-Boot wird der Masterboot vom Sektor 0 der Platte geladen.
2. Der Masterboot lädt dann den Partition-Boot-Block vom Sektor 0 der aktiven Partition.
3. Anschließend wird der Rest des Boot's von den nächsten 29 Sektoren der Platte geladen.

Beim ersten Start bringt der Boot folgende Statusmeldung :

Booting the UNIX System ...

Die Anweisungen für den Boot sind als Defaultwerte in der Datei /stand/boot hinterlegt. Mit RETURN werden die Defaultwerte genutzt. Wenn eine andere Taste benutzt wird, wartet der Boot auf eine Anwenderanweisung.

siehe Kommando
boot (1M)

1.3 init

Kommando : /sbin/init cmd
└── run-level(0123456SsQqabc)

Datei : /etc/inittab
 /var/adm/utmp user- und accounting-Informationen
 für die Kommandos who,write, login.
 /var/adm/wtmp Historie von user-Logins.

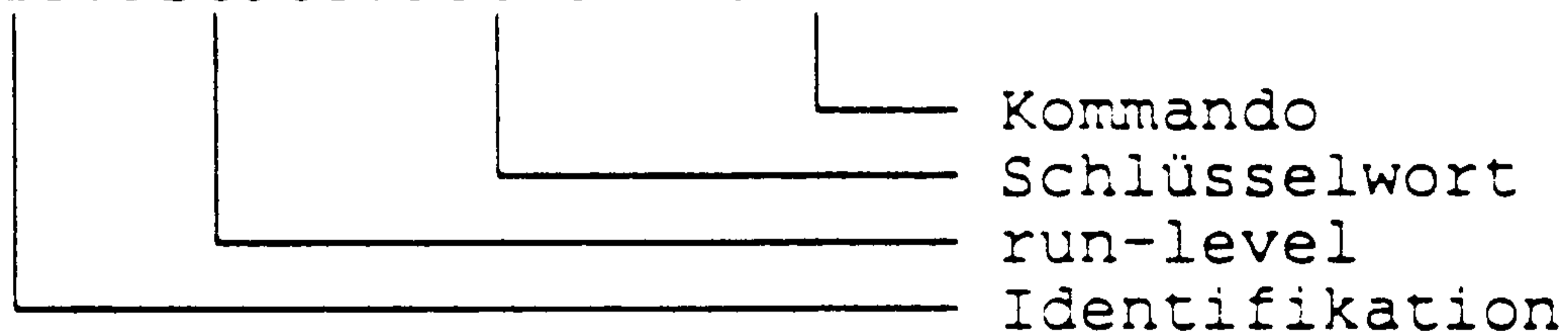
1.3.1 inittab

Jose d. IPL

Der Prozess "init" durchläuft die "inittab" zyklisch von oben nach unten. In Abhängigkeit vom run-level werden die Einträge ausgeführt.

Datei : /etc/inittab

Format : id:rstate:action:command



1.3.1.1 Identifikation

id - ist einzig zur Identifikation der Einträge, d.h. Kommentar für Benutzer.

1.3.1.2 Run-Level

rstate - definiert den run-level bei dem dieses Kommando ausgeführt wird.

0 - Poweroff

- 1 - Single User : das Multi-User-System kann genutzt werden jedoch kann man sich nur am tty000 anmelden. Alle Prozesse des Multi-User-Systems sind gestartet und alle Kommandos stehen zur Verfügung (Prozesse im Anhang)
- s,S - Single User : keine Multi-User-Funktionen möglich d.h. man kann sich nur an der Konsole einloggen. Es stehen keine Multi-User-Kommandos zur Verfügung (Prozesse im Anhang)
- 2 - Multi-User-System.
- 3 - Multi-User-System mit automatischem Start vom DFS/NFS.
- 4 - Benutzer Systemstart : nicht benutzt.
- 5 - Halt und Reboot. (Der in den AT&T-Unterlagen beschriebene Firmware-Monitor funktioniert nur bei originalen AT&T Maschinen)
- 6 - Halt und Reboot.
- q,Q - nochmaliges Ausführen der "inittab". (wie bei run-levelstart) Wird gestartet wenn ein neues Terminal eingetragen wurde.
- a,b,c - Pseudostatus ; die mit a,b oder c gekennzeichneten Einträge werden ausgeführt ohne den Systemstatus zu verändern. (Start von Anwendungen).

Hinweis : Es können bei einem Eintrag mehrere Run-Levels eingetragen werden (siehe Beispiel inittab).

1.3.1.3 Schlüsselwort

action - Das Schlüsselwort teilt dem "init" mit, wie der Prozess im Kommandofeld zu behandeln ist.

respawn - Startet den Prozess wenn er nicht vorhanden ist. Startet den Prozess erneut, wenn er stirbt. Geht in der Bearbeitung der inittab weiter ohne auf eine Endmeldung zu warten (z.B. getty).

- wait - Startet den Prozess und wartet auf seine Beendigung.
- once - Startet den Prozess und wartet nicht auf eine Endmeldung. Stirbt der Prozess, wird er nicht wieder gestartet. Gleiches gilt bei Änderung des run-levels.
- boot - Startet den Prozess nur während der "init" Startphase (erster Durchlauf der inittab) und wartet nicht auf eine Endmeldung. Stirbt der Prozess, wird er nicht wieder gestartet.
- bootwait - Startet den Prozess nur beim ersten Übergang vom Single-User- auf den Multi-User-Betrieb und wartet auf eine Endmeldung. Stirbt der Prozess, wird er nicht wieder gestartet.
- powerfail - Startet den Prozess nur beim power fail Signal (SIGPWR). Wartet nicht auf eine Endmeldung
- powerwait - Startet den Prozess nur beim power fail Signal (SIGPWR) und wartet auf eine Endmeldung.
- off - Wenn ein Prozess gestartet ist wird ein Warnsignal (SIGTERM) gesendet und 5 Sekunden gewartet, bevor der Prozess mit dem kill-Signal (SIGKILL) beendet wird (z.B. Teleservice)
- ondemand - Synonym für "respawn". Wird nur bei "rstate" a,b oder c gestartet
- initdefault - Dieser Eintrag wird nur bei erstmaligem Durchlauf der "inittab" aufgesucht. Er bestimmt den "run-level", mit dem Hochfahren wird. Fehlt ein Eintrag erscheint beim Hochlauf eine Abfrage nach dem "run-level".
- sysinit - Wird nur ausgeführt bevor die Konsole initialisiert ist, d.h. bevor eine Eingabe möglich ist.

1.3.1.4 Kommandos

command - ist das auszuführende Kommando. Hier kann jedes shell-Kommando oder Programm angegeben werden.

Who -R = dove mi ha dato in che livello

Esempio del 4/3/13 pg

1.4 Beispiel einer "inittab"

esime due copie sotto /etc/conf/cf.d/init.000

```
cc::sysinit:/sbin/chkconsole > /dev/sysmsg 2>&1
ap::sysinit:/sbin/autopush -f /etc/ap/chan.ap
ck::sysinit:/sbin/setclk </dev/console >/dev/sysmsg 2>&1
bchk::sysinit:/sbin/bcheckrc </dev/console >/dev/sysmsg 2>&1
is:2:initdefault:
r0:0:wait:/sbin/rc0 off 1> /dev/sysmsg 2>&1 </dev/console
r1:1:wait:/sbin/rc1 1> /dev/sysmsg 2>&1 </dev/console
r2:23:wait:/sbin/rc2 1> /dev/sysmsg 2>&1 </dev/console
r3:3:wait:/sbin/rc3 1> /dev/sysmsg 2>&1 </dev/console
r5:5:wait:/sbin/rc0 reboot 1> /dev/sysmsg 2>&1 </dev/console
r6:6:wait:/sbin/rc6 reboot 1> /dev/sysmsg 2>&1 </dev/console
li:23:wait:/usr/bin/lm /dev/systty /dev/syscon >/dev/null 2>&1
sc:234:respawn:/usr/lib/saf/sac -t 300
sc0:23:off:/sbin/getty tty0 S
sc1:23:off:/sbin/getty tty1 S
s000:12345:respawn:/sbin/getty tty000 S
s001:23:respawn:/sbin/getty tty001 S
s003:23:respawn:/sbin/getty tty003 S
s006:23:respawn:/sbin/getty tty006 S
s007:23:respawn:/sbin/getty tty007 S
s011:23:respawn:/sbin/getty tty011 S
```

→ STARTA IL PROCESSO quando non lo fare

1.4.1 /sbin/chkconsole

Führt in der Initialisierungsphase einen HW-Check auf die Konsole durch (Terminaltyp der Konsole).

Kommando : /sbin/chkconsole

1.4.2 /sbin/autopush

Dieses Kommando erlaubt es einem eine Liste von Modulen zu erstellen, die automatisch durchlaufen wird, wenn das Device geöffnet wird.

Kommando : autopush -f file

-f - Diese Option gibt das File für die
autopush-Konfiguration der Treiber an.

Beispiel : /sbin/autopush -f /etc/ap/chan.ap
↳ siehe Kapitel 2

1.4.3 /sbin/setclk

Dieses Kommando initialisiert die Systemuhr mit Hilfe der
Hardwareuhr.

Kommando : /sbin/setclk

1.4.4 /sbin/bcheckrc

Shell-Prozedur, die beim Systemstart die Dateisysteme
überprüft und ein mount auf /proc , /dev/fd und /var
durchführt.

Prozedur : /sbin/bcheckrc

Datei : /etc/vfstab Dateisystemtabelle
 /etc/mnttab Mounttabelle

siehe Anhang

1.4.5 /sbin/rc? - Prozeduren

Je nach angegebenen run-level werden verschiedene rc-
Prozeduren gestartet.

run-level	prozedur	directory
0	/sbin/rc0	/etc/rc0.d
1	/sbin/rc1	/etc/rc1.d
2	/sbin/rc2	/etc/rc2.d

3	/sbin/rc2 & /sbin/rc3	1. /etc/rc2.d 2. /etc/rc3.d
5	/sbin/rc0	/etc/rc0.d
6	/sbin/rc6	/etc/rc0.d

Die rc-Prozeduren suchen die shell-Prozeduren unter den korrespondierenden DVZ's und führen sie je nach Zustand (Start/Stop) aus.

Die Prozedurnamen sind folgendermaßen aufgebaut:

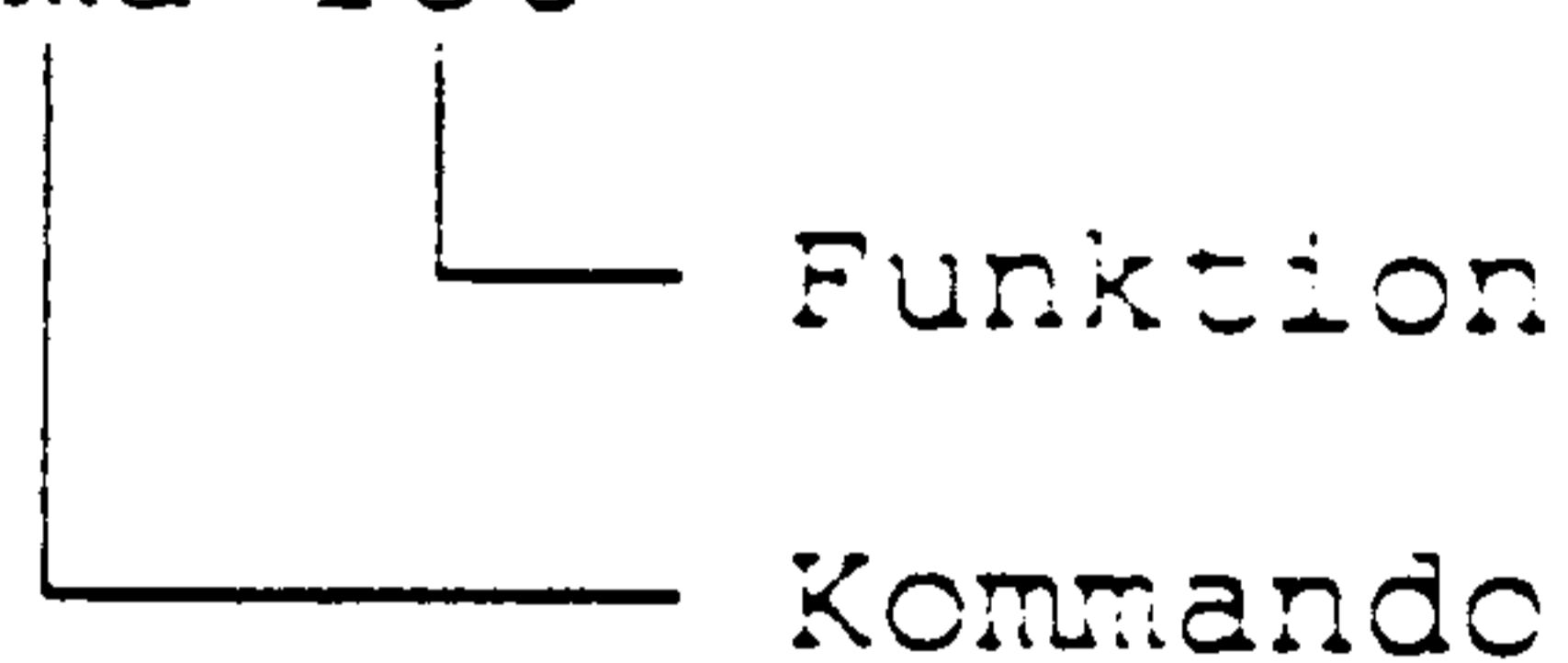
fase di comando
 S* - für Startprozeduren
 K* - für Stopprozeduren
fase di chiusura

siehe Anhang

1.4.6 /sbin/uadmin

Dieses Kommando dient zum Ändern des Systemstatus (run-level).

Kommando : /sbin/uadmin cmd fct



cmd - 1 = reboot

2 = shutdown

fct - 0 = halt poweroff

1 = reboot von /stand/unix

2 = interaktiver reboot

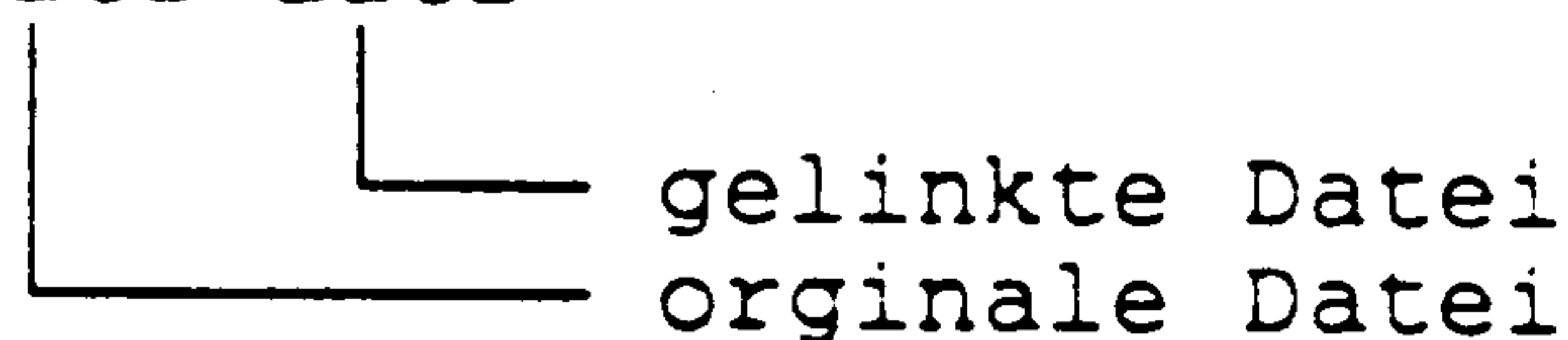
3 = reboot

2 file di link = ln =

1.4.7 /usr/bin/ln

Erzeugt einen Hardlink von Dateien

Kommando : /usr/bin/ln dat1 dat2



1.4.8 /usr/lib/saf/sac

Der Service Access Controller ist ein Kontrollprogramm für Servermaschinen. Der SAC überwacht Portmonitore, unter anderem auch den getty.

Die Aufgaben des SAC können vom Systemadministrator spezifiziert werden. Mögliche Aufgaben sind :

Überwachen des Portstatus
Starten/Stoppen von Monitoren

Kommando : /usr/lib/saf/sac -t time

└───┬───┘
└───┬───┘ Pollingzeit zur Abnahme
der Statusmeldungen.

1.4.9 /etc/getty

Kommando : /sbin/getty dev name

└───┬───┘
└───┬───┘ Eintrag in /etc/gettydefs
Gerätedateiname

Datei : /etc/gettydefs

siehe Anhang

Beispiel : Der Standard-Eintrag für SIEMENS-7-Bit-Terminal

```
S# B38400 OPOST ONLCR TAB3 IGNPAR IXON ISTRIP ECJO ECHOE
ECHOK ICANON ISIG CS7 CREAD PARENB PARODD # B38400 OPOST
ONLCR TAB3 IGNPAR IXON ISTRIP ECJO ECHOE ECHHOK ICANON
ISIG CS7 CREAD PARENB PARODD # login : #S
```

Informationen finden Sie im CES-Manual Kapitel 4.

diskussion diskop

2. Dateisystemtypen

2.1 Platteneinteilung

zusätzlichem SLICE (0 Partitionen)

Beispiel einer Plattenbelegung :

Partition	Belegung	Dateisystem	
0	boot	-	Pflicht
1	root	s5 ufs	
2	swap	-	
3	usr	s5 oder ufs	
4	-	-	
5	-	-	
6	-	-	
7	-	-	
8	-	-	
9	-	-	
10	stand	bfs	
11	var	s5 oder ufs	
12	home12	s5 oder ufs	
13	tmp	s5 oder ufs	
14	home14	s5 oder ufs	
15	-	-	

Die Belegung der Partitions ist wahlfrei. Sie wird bei der Systeminstallation festgelegt.

2.2 s5 - System

Das s5 - Dateisystem besteht zur Adressierung des Plattenbereichs aus 4 Kategorien von Blöcken.

Block 0	Boot-Block
Block 1	Super-Block
Block 2	Inodes
Block n	
Block n+1	Daten-Blöcke
.	
Ende des Dateisystems	

s5 - Boot-Block

Obwohl der Boot-Block ein Teil des Dateisystems ist, wird er nicht vom Dateisystem benötigt. Er ist reserviert für Prozeduren die zum Booten des Systems genutzt werden. Wenn das Dateisystem nicht zum Booten benutzt wird, bleibt der Boot-Block ebenfalls ungenutzt.

s5 - Super-Block

Viele Informationen über das Dateisystem werden im Super-Block hinterlegt. Dazu gehören :

1) Dateisystem und Status

- Label (Dateisystemname)
- Größe in logischen Blöcken
- Read-Only Flag
- Datum und Uhrzeit des letzten Updates

2) Inodes

- Nummern aller zugewiesenen Inodes
- Nummern der freien Inodes
- Gruppen von jeweils 100 freien Inodes
- Einen Index für Gruppen von freien Inodes

3) Datenblöcke

- alle Nummern der freien Blöcke
- Gruppen von jeweils 50 freien Nummern
- Einen Index für Gruppen von freien Nummern

s5 - Inodes

Eine Liste von Inodes wird eine i-Liste und die Position der Inodes in der i-Liste wird i-Nummer genannt.

Ein Inode enthält alle Verwaltungsinformationen einer Datei, jedoch keinen Dateinamen. Der Dateiname wird im Dateiverzeichnis hinterlegt.

Ein Inode besteht aus 64 Bytes. Jeweils 8 Inodes bilden einen physikalischen Block (Blockgröße 512 Byte). Die Länge der i-Liste ist nicht fest. Sie wird beim erstellen des Dateisystems festgelegt.

Ein s5-Inode enthält :

- den Typ einer Datei
 - "-" reguläre Datei
 - "d" directory
 - "b" Block-Device
 - "c" Character-Device
 - "l" Symbolischer Link
 - "p" FIFO; named pipe
 - "s" Socket
- den Modus einer Datei
 - "r" read
 - "w" write
 - "x" execute
- Dateigröße in Byte

- Anzahl der Hardlinks
- USER-ID des Eigentümers
- Gruppen-ID zu der Datei
- 13 Einträge für die Adressen der Plattenblöcke
- Datum und Uhrzeit des letzten Zugriffs
- Datum und Uhrzeit der letzten Änderung
- Datum und Uhrzeit der Dateierstellung

Die Einträge der Adressen für die Plattenblöcke sind das Herz der Inodes. Die ersten 10 Adressen zeigen direkt auf Plattenblöcke, der 11te Eintrag zeigt indirekt, der 12te Eintrag doppelt indirekt und der 13te Eintrag dreifach indirekt auf die Plattenblöcke.

Durch die Verkettung der Adressen eines Inodes ergibt sich ein maximal adressierbarer Datenbereich :

Blockgröße 512 Byte -> 1 GByte
Blockgröße 1024 Byte -> 16 GByte
Blockgröße 2048 Byte -> 256 GByte

Die maximale Größe einer Datei bestimmt sich aus dem 4Byte großem Dateigrößeneintrag in der Inode (max. 4 GByte)
Die maximale Inodeanzahl beträgt 65532 Einträge

s5 - Daten-Blöcke

Der Rest des allokierten Speichers des Slices werden als Datenblöcke benutzt.

Eine normale Datei umfasst komplette Datenblöcke. Bei Dateiverzeichnissen besteht der Datenblock aus 16 Byte großen Einträgen. Jeder Eintrag stellt einen Dateinamen oder ein Unterverzeichnis dar. 2 Byte eines Eintrags werden für die i-Nummern und 14 Byte für den Namen genutzt.

Achtung : Die maximale Dateinamenlänge beträgt 14 Zeichen !

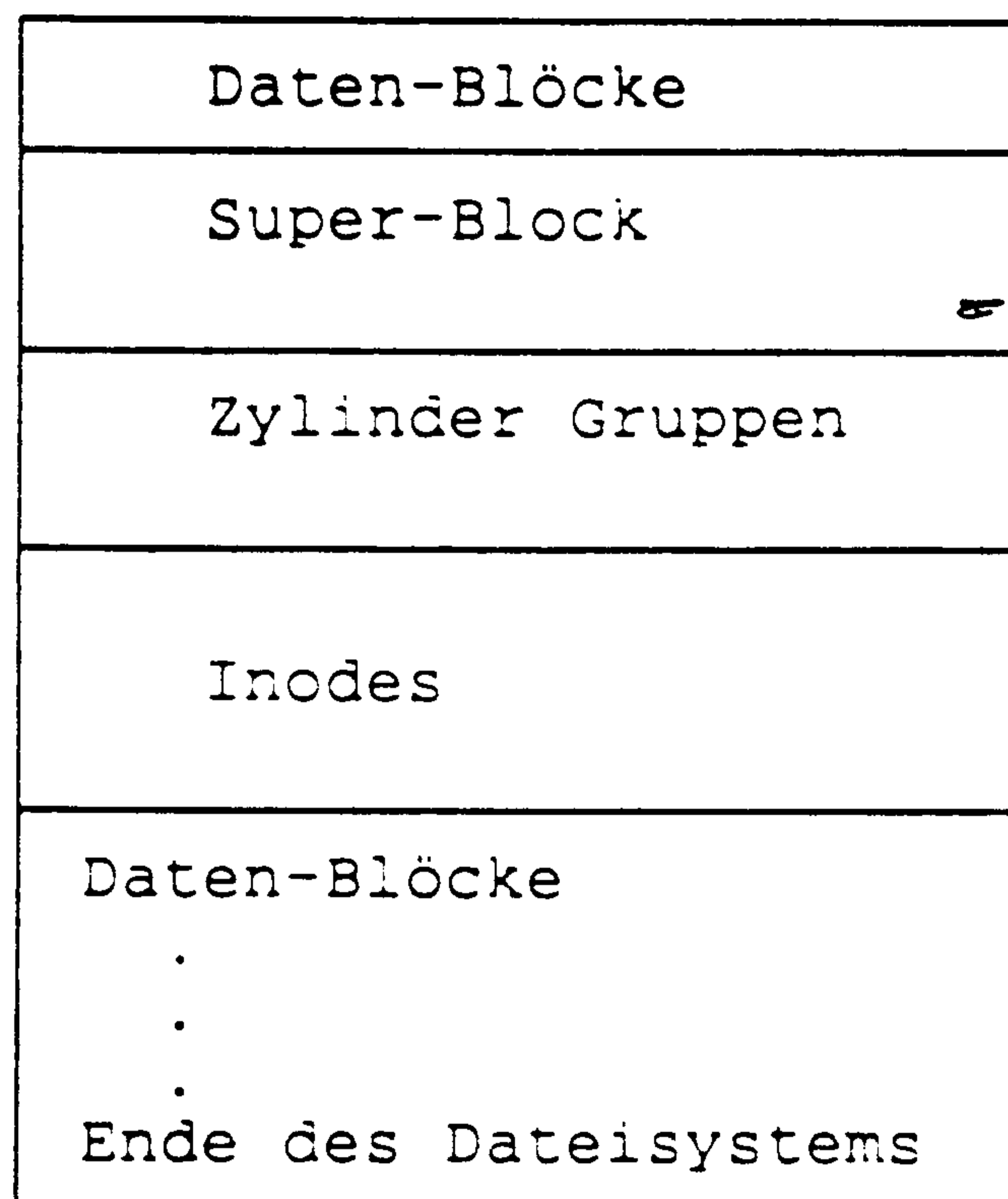
s5 - Freie Blöcke

Die noch nicht genutzten Blöcke werden in einer Kette von Blöcken aneinandergelinkt. Jeder Block enthält die Adresse des nächsten freien Blocks.

2.3 ufs - System

Das ufs-Dateisystem (Berkley Fast Filesystem) ist wesentlich komplexer aufgebaut als das s5-Dateisystem. Im Vergleich zum s5-Dateisystem gibt es Zylindergruppeninformationen, die jeweils eine Kopie des Super-Blocks, eine eigene Inode Liste, und weitere statistische Informationen enthalten.

Zylinder Gruppe 1



Dies hat folgende Vorteile :

Höhere Datensicherheit des Dateisystems
Höhere Performance

Die höhere Datensicherheit wird erreicht durch :

- ein dynamisches Verschieben der Metadaten (Super-Block, Inode Liste, etc.)
Daraus resultiert, daß der Super-Block immer auf einer anderen Oberfläche liegt.

Die höhere Performance wird erreicht durch :

- Anlegen von Dateien innerhalb weniger Zylinder
- Große File Block Size (8KB)
- Bitmaps statt verketteter Listen für die Verwaltung der freien Blöcke

ufs - Boot-Block

Der ufs - Boot-Block ist ausschließlich auf der ersten Zylindergruppe und belegt die ersten 8KB auf der Partition. Dieser Bereich ist reserviert für Prozeduren zum Booten des Systems. Wird der Boot-Bereich nicht benötigt, wird er mit Blanks aufgefüllt.

ufs - Super-Block

Der Super-Block enthält Informationen, die den Aufbau des Dateisystems beschreiben :

- Anzahl und Größe der Zylindergruppen
- Zylindergruppen Summen Informationen
- Block und Fragmentgröße
- Anzahl der Spuren, Sektoren, Inodes und Zylinder pro Zylindergruppe

Um einen hohen Verschnitt durch die große Blockgröße (8KB) zu vermeiden, wird jeder Block fragmentiert. Mögliche Fragmentgrößen sind 512, 1024, 2048 und 4096. Bei Sinix V5.4 wird eine Fragmentgröße von 1024 genutzt. Die Festlegung der Fragmentgröße erfolgt mit dem Kommando "mkfs" beim erstellen des Dateisystems.

ufs - Inodes

Der Inode ist das zentrale Element des Dateizugriffs. Er enthält Daten für :

- den Typ einer Datei
- den Link Zähler
- die Dateigröße
- Modifikations-, Zugriffs- und Erstellungszeitpunkt
- Zugriffsrechte
- Eigentümer und Gruppenzugehörigkeit
- Blockadressen für die ersten 12 direkt adressierten Blöcke
- Blockadressen für einfach, zweifach und dreifach indirekte Adressierung

Ein Inodeeintrag besteht aus 128 Byte.

ufs - Zylindergruppen

In den Zylindergruppen sind Informationen über :

- Bitmaps für Inodes und freie Blöcke
- Zylindersummeninformationen
- Anzahl der Zylinder, Inodes und Blöcke
- Anzahl und Position der freien Blöcke auf den Zylindern
- Position des zuletzt benutzten Blocks, Fragments und Inodes
- Zeitpunkt an dem die Zylindergruppeninformation das letzte mal auf die Platte geschrieben wurde

ufs - Daten-Blöcke

Die Reste des angewiesenen Speichers auf dem Slice werden als Datenblöcke und Fragmente benutzt. Eine normale Datei umfaßt komplette Fragmente. Bei Dateiverzeichnissen besteht der Datenblock aus 255 Byte großen Einträgen. Jeder Eintrag wird für den Dateinamen oder einen Unterverzeichnisnamen und die i-Nummer genutzt.

ufs - Freie Blöcke

Die noch nicht genutzten Blöcke und Fragmente werden als freie Blöcke und Fragmente in der Zylindergruppen-Summen-Information gekennzeichnet.

2.4 bfs - System (boot FS)

Das bfs - Dateisystem ist ein spezielles Dateisystem für stand-alone Programme (z.B Unix) und alle Textdateien für die Boot-Prozeduren.

Das bfs - Dateisystem erlaubt ein schnelles und einfaches Booten. Das einzige Dateiverzeichnis das bekannt ist, ist das "root"-Dateiverzeichnis. Es ist einem Benutzer nur möglich, normale Dateien anzulegen. Spezielle Dateien oder Dateiverzeichnisse können nicht angelegt werden.

Es besteht aus drei Teilen :

Super-Block
Inodes
Daten-Blöcke . . . Ende des Dateisystems

bfs - Super-Block

Im Super-Block sind folgende Informationen hinterlegt :

- "magic number"; notwendig zur Identifikation des Dateiverzeichnistypes
- die Größe des Dateisystems
 - Startpunkt der Datenblöcke in Byte
 - Endpunkt der Datenblöcke in Byte
- "sanity words". Diese 4 Worte werden vom fsck nach einem Systemcrash zum Überprüfen des Dateiverzeichnisses benötigt.

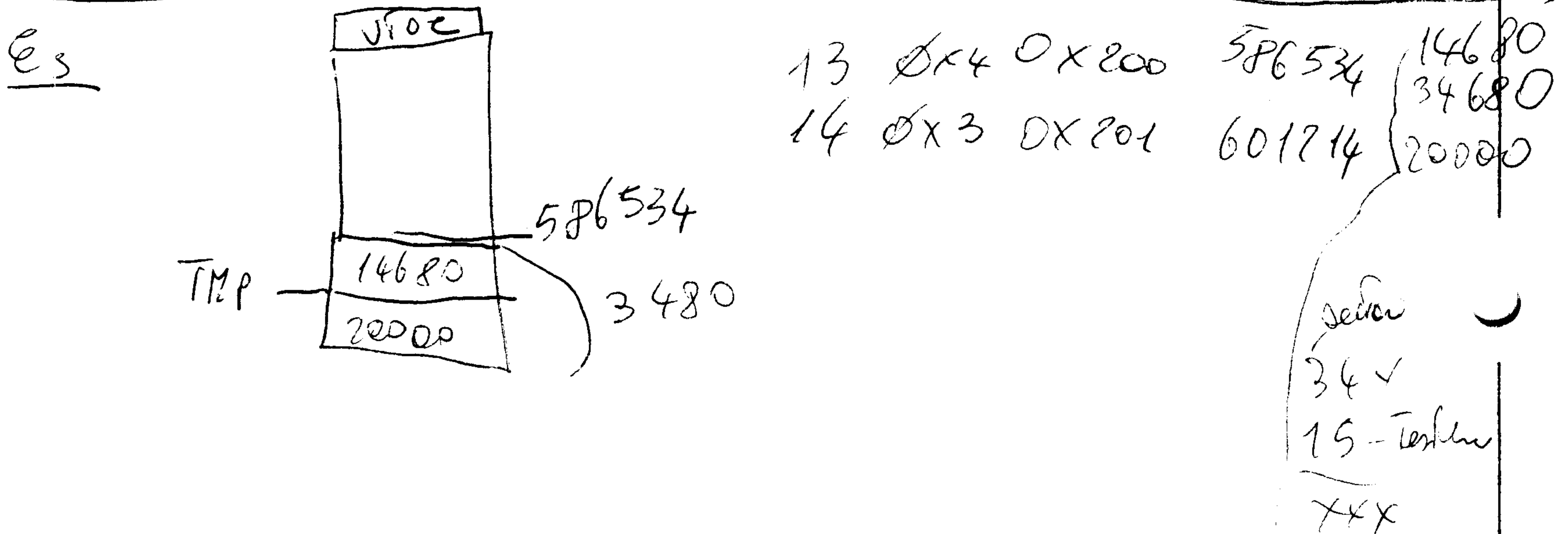
bfs - Inodes

Ein Inode enthält alle Informationen der Dateien außer dem Dateinamen. Die Dateinamen sind im "root"-Verzeichnis, dem einzigen Dateiverzeichnis, das das bfs-Dateisystem kennt, hinterlegt. Ein Inode besteht aus 64 Byte und enthält folgende Informationen :

- die Inodenummer
- den ersten Datenblock
- den letzten Datenblock
- das Ende des Dateisystems in Byte
- die Dateiattribute
- Typ und Modus der Dateien
- Benutzer- und Gruppen-ID der Datei
- Hardlinks
- Modifikations-, Zugriffs- und Erstellungszeitpunkt

bfs - Datenblöcke

Die Größe der Datenblöcke ist 512 Byte. Sie werden genutzt um das "root"-Verzeichnis und normale Dateien zu speichern. Eine normale Datei nutzt den gesamten Block, während das "root"-Verzeichnis aus Einträgen von jeweils 16 Byte besteht. Jeder Eintrag setzt sich aus 2 Byte i-Nummer und 14 Byte Namen zusammen.



Swap - slice non montabile

(INTEL) Slice Ø = Partitione Ø (in NATIONAL)

le maxime occupation d' disco intel = partitione & grosse national

X volume UTOE

- 1) Single user
 - 2) Soluzione dati min = partitione
 - 3) Lettura PRVTOE-U-F file slice Ø
 - 4) Umanità / TMP
 - 5) editore di file
 - 6) oo(Utoe - j file slice Ø
 - 7) mkfs U-F vjs deve dimensione
- base file user

1) Platten in SINIX V5.4

Bootprogramm

COMMAND

**disksetup -b /install/etc/boot
/dev/rdisk/c0d0s0**

Im Minisystem wird Bootprogramm auf Label der System-Platte geschrieben.

physikalische Platteneinteilung

fdisk /dev/rdisk/cxdys0

x = Controlemnummer
y = Plattennummer

physikalische Platteneinteilung, wobei nur eine Partition aktiv sein kann

programmazione di settore e formata slice

Slice-Platteneinteilung + FS + vfstab + Mount-DVZ

disksetup -I /dev/rdisk/cxdys0

CREA SLICE

subdividere slice

Slices werden in der aktiven Partition einer **normalen** Platte eingerichtet :

Anzahl der Slices
Größe der Slices
Filesystemtyp
Name der Mount-DVZs

disksetup -d lay -IB /dev/rdisk/cxdys0

lay = selbstdefinierte Tabelle :

10	/stand	bfs	5M
2	/dev/swap	-	2m
1	/	ufs	31M
3	/usr	ufs	60M
4	/home	ufs	45W
11	/var	ufs	60W
12	/opt	ufs	45W
13	/tmp	ufs	10W

slice	Name	FS-Typ	Größe
1-14	Mount-DVZ	na	M=Mb
		s5	m=Memory
		ufs	W=% Rest

Im Minisystem werden Slices in der aktiven Partition einer **System-Platte** eingerichtet :

Anzahl der Slices
Größe der Slices
Filesystemtyp
Name der Mount-DVZs

(Root-FS ist an /mnt des Minisystems)

Anmerkung :

Falls nur die Slice-Einteilung defekt ist, gibt man in dieser Tabelle für den Platten-Typ na ein (na = no access). Dann werden keine neuen Filesysteme angelegt. Kein Datenverlust !!!

PROGRAMMI TRATTAMENTO DEI DISCHI delle **VTOC** (volume partition table)

prtvtoc /dev/rdisk/cxdys0

OPZIONI

prtvtoc -e /dev/rdisk/cxdys0

lo scrive sul file

prtvtoc -f aaa /dev/rdisk/cxdys0

Filei aaa editieren

edvtoc -f aaa /dev/rdisk/cxdys0

*perché di scrivere la info della VTOC
sulle o avere le dimensioni slice*

legge la tabella del disco
Anzeigen der Platteneinteilung (Slices)
(vtoc = virtuell table of contents)

lo scrive in un file
Platteneinteilung (Slices) wird in die Datei /etc/partitions geschrieben.

FILE LEGGIBILE

Platteneinteilung (Slices) in eine Datei schreiben (z.B. aaa). Diese Datei kann verändert werden. Die veränderte Slice-Einteilung wird in den VTOC geschrieben.

2) Platten in SINIX V5.4

Slice-Platteneinteilung

mkpart -i device

VTOC löschen *HA + OPTIONAL*
VEDI COMMAND

mkpart -p slicename device

Slice entfernen aus VTOC

mkpart -P slicename device

Slice hinzufuegen in VTOC

Seifried → /etc/partitions :
slicename = z.B.: home12
device = z.B.: disk10

*merce n' avoir coller
lent avec DS
une TAG de disque
case e*

Swap-Slices

swap -l

o place

Informationen über den Swap-Bereich
informations sulle ore di SWAPP

swap -a /dev/dsk/cxdysz 0 size

Aktivieren eines weiteren Swap-Slices

x = Controlemummer

y = Plattennummer

z = Slicenummer

size = Blockgröße

attivare uno slice ore di SWAPP

swap -d /dev/dsk/cxdysz 0

Deaktivieren eines Swap-Slices

Filesystem-Typ

fstyp /dev/rdisk/cxdysz

Gibt Filesystemtyp aus : ufs ; s5

fstyp -v /dev/rdisk/cxdysz

Gibt Filesystem-Parameter aus.

Filesystem einrichten, reparieren, mounten

mkfs -F typ /dev/dsk/cxdysz size

Filesysteme des angegebenen Typs werden eingerichtet

typ = FS-Typ (ufs oder s5)

size = Größe (siehe /etc/partitions)

fsck -F typ /dev/dsk/cxdysz

Filesysteme des angegebenen Typs werden gecheckt

mount -F typ /dev/dsk/cxdysz dvz

Filesysteme werden gemountet

dvz = Mount-DVZ

Filesystem-Belegung

df -k

Anzeigen des freien/belegten Platzes pro Filesystem

dfspace

Anzeigen des freien Platz pro Filesystem

Minisystem V5.4

Minisystem laden

SINIX0 : Floppy einlegen + MX300 einschalten

Aufforderung : SINIX1

SINIX1 : Kernel wird geladen

File-System s5

FS ins Ram-Disk

INSTALL

Aufforderung : SINIX2

SINIX2 :

File-System s5

mount /dev/fd0 /install

Tastaturabfrage

INSTALLA

Beenden der Installation : DEL

Diagnose der Systemplatte

1. Mounten der Filesysteme

```
/etc/fs/s5/fscck -y /dev/rdisk/c0d0s1      root-Filesystem
/etc/fs/s5/mount /dev/dsk/c0d0s1 /mnt
/etc/fs/ufs/fscck -y /dev/rdisk/c0d0s3     usr-Filesystem
/etc/fs/ufs/mount /dev/dsk/c0d0s3 /mnt/usr
```

2. Shell-Variablen setzen

```
PATH=$PATH:./mnt/bin:/mnt/sbin:/mnt/usr/sbin:/mnt/usr/bin
USER=root
TERM=97801
export PATH USER TERM
```

Handwritten: TERMINFO = /mnt/usr/share/lib/terminfo

3. Besonderheit

```
ced :
mkdir usr/share
mkdir /usr/share/lib
ln -s /mnt/usr/share/lib/terminfo /usr/share/lib/terminfo
```

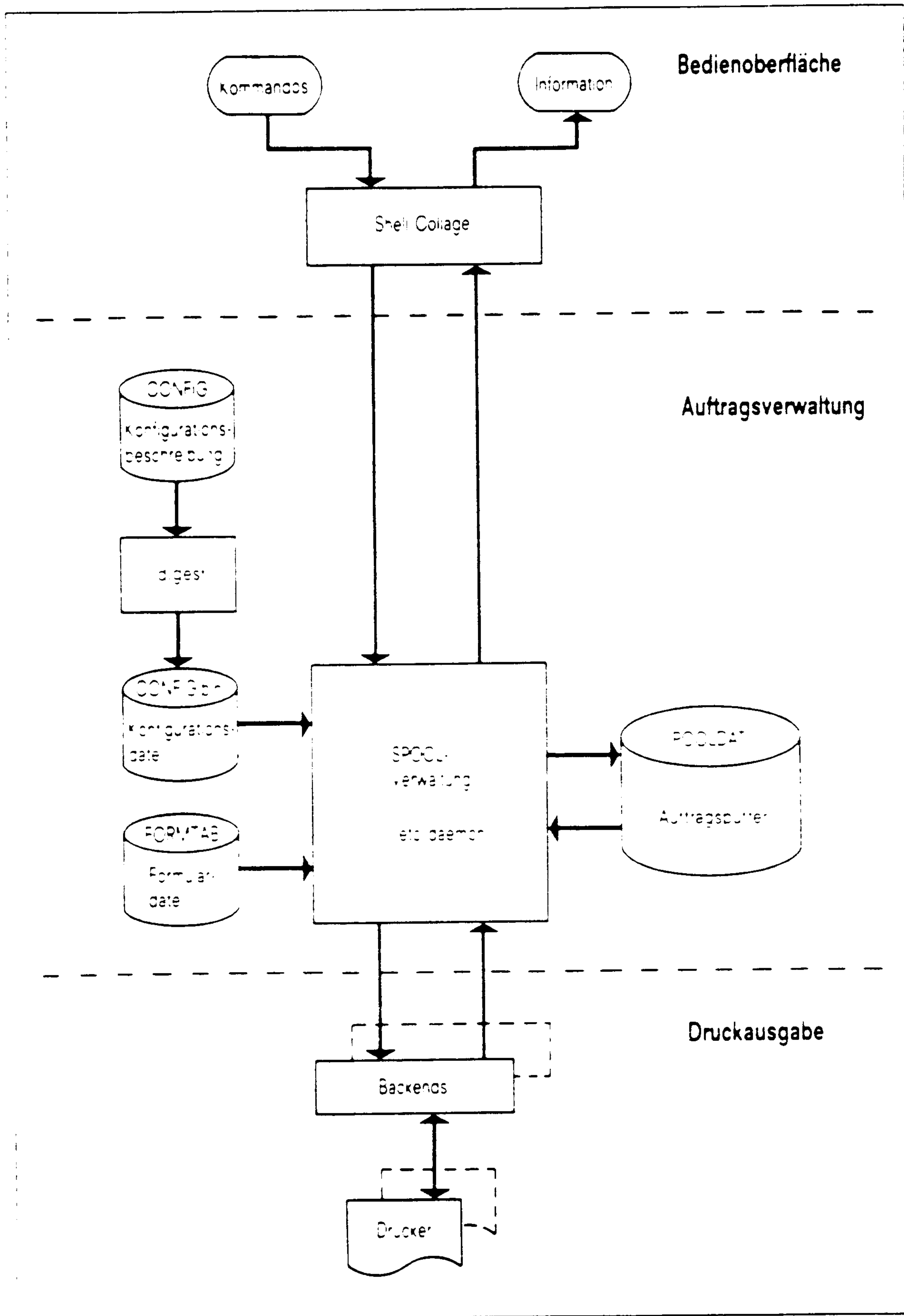
OSCITA DAL MINISISTEMA

Beenden des Minisystems

```
cd /
sync
uadmin 2 0      ( shutdown )
```

SINIX-SPOOL-V3.0

SPOOL = Simultaneous Peripheral Operations On Line



SINIX-SPOOL-V3.0

Die Konfigurationsdatei CONFIG

Die Konfigurationsdatei enthält in lesbarer Form:

- Welche Drucker das SPOOL-System verwaltet.
- Welche Zuordnung zwischen Druckernamen, Backends und Gerätedateien besteht.
- Welche druckerspezifischen Parameter des Kommandos lpr von .../daemon an die Backends weitergeleitet werden.
- Welche Drucker zu einer Druckergruppe zusammengefaßt sind.
- Welche Benutzer Druckerverwalter sind, d.h. privilegierte Funktionen für bestimmte Drucker ausführen können.
- Welche Benutzer und Terminals welchen Druckergruppen zugeordnet sind.

Die CONFIG-Datei wird bei der Konfigurierung über das SINIX-Bediensystem automatisch angelegt bzw. verändert.

Sie besteht aus vier Teilen:

- 1. Teil: Angabe der konfigurierten Drucker
- 2. Teil: Angabe der definierten Druckergruppen
- 3. Teil: Angabe der autorisierten Druckerverwalter
- 4. Teil: Zuordnung der Benutzer/Terminals zur Druckergruppe

Beispiel für eine Datei .../CONFIG

```
copy etc
D003 '/usr/spool/spooler/lp9013' /dev/tty003 -zs= -dt -int
      -pb1 -pb2 -pb3 -pb= -pl= -hd -hd -trl -trl -font= -ab= -bis=
D007 '/usr/spool/spooler/lp9001 -pb2 -dt' /dev/tty007 -zs= -dt -int
      -zb= -pb1 -pb2 -pb3 -hd -hd -trl -trl -ab= -bis=
D008 '/usr/spool/spooler/lp9047 -pl=60' /dev/tty008 -pb= -pl= -hd
      -hd -trl -trl -ab= -bis= -tst -band=
DF '/usr/spool/spooler/interface -prog=/usr/xaver/backend -cbreak
    -odd even -speed=89600' /dev/tty002 -dt -pb2
SPAZIO
ALLE ( D003 D008 D007 DF ) 'Alle vorhandenen Drucker'
G003 ( D003 ) 'Multifunktionsdrucker 9013 mit Einzelblattzufuhr D003'
G007 ( D007 ) 'Nadeldrucker 9001 D07'
G008 ( D008 ) 'Banddrucker 9047 D008'
GS   ( D003 D007 D008 ) 'Alle schnellen Drucker D003, D007, D008'
GF   ( DF   ) 'Fremddrucker, ueber Privatbackend angesprochen DF'

admin ( D003 D007 D008 DF )
franz ( D003 )
xaver ( D007 DF )
nicole ( D008 )

/dev/tty001 G007
franz      GS
```

SINIX-SPOOL-V3.0

Das Interface-Backend

Um den Anschluß weiterer Fremddruckern zu erleichtern, wird ein Interface-Backend ... *interface* ausgeliefert, das zur Anbindung von einfachen privaten Backend-Programmen dient. Das Interface-Backend übernimmt die Kommunikation mit dem daemon und ruft das private Backend auf, das die Ausgabe einer Datei auf den Drucker übernehmen muß. Die Standard-Ein-/Ausgabe der Backends werden bereits vom daemon auf die Gerätedatei des Druckers umgelenkt.

Diese einfachen privaten Backends müssen daher lediglich eine Datei, deren Name als Parameter übergeben wird, auf ihre Standard-Ausgabe kopieren und eventuell zuvor einige druckspezifische lpr-Schalter (z.B. *-pb2* oder *-dt*) auswerten. lpr-Schalter, die nicht druckerspezifisch sind (z.B. *-nc =*), werden bereits vom Interface-Backend oder vom daemon ausgewertet.

Eventuelle Rückmeldungen des Druckers können auch von diesen einfachen Backends aus der Standard-Eingabe gelesen werden.

Die Programme */bin/cat* oder */bin/pr* können als universelle primitive Backends mit Hilfe des Interface-Backends in die Druckerverwaltung integriert werden.

Im Gegensatz zu den "echten" Backends wird die Bereitschaft des Druckers vom Interface-Backend nicht überprüft. Der Drucker wird immer als BEREIT vorausgesetzt und es wird lediglich das Datenflußprotokoll XON/XOFF verwendet.

Falls ein neuer Druckertyp über das Interface-Backend betrieben werden soll, so kann der Systemverwalter den Druckaufruf in die Datei */usr/admin/.colface/Konf/dtype* eintragen, um ihn anschließend ganz normal über das Bediensystem konfigurieren zu können.

PARAMETRI DI INTERFACCIA Bedeutung der angegebenen Parameter für das Interface-Backend

-prog = PROGRAMMNAME

hier kann der absolute Pfadname des eigenen Backends angegeben werden.

+ odd / -odd

ungerade Parität setzen / wegnehmen

+ even / -even

gerade Parität setzen / wegnehmen

Falls kein Paritätsbit gewünscht wird, so sollte *+ odd + even* gesetzt werden.

+ cbreak

für Druckerausgabe sollte immer der CBREAK-Modus verwendet werden.

+ crmod / -crmod

im CRMOD sorgt der Treiber dafür, daß vor einem Zeilenvorschub auch ein Wagenrücklauf durchgeführt wird.

-speed =

Einstellung der Übertragungsrate

-bits =

Einstellung der Datenbits pro Zeichen

+ rts

Einschalten der Hardware-Datenflußsteuerung via CTS/RTS
(nur gültig für TAKSI-Verbindungen)

-rts

Einschalten der Software-Datenflußsteuerung via XON/XOFF
(nur gültig für TAKSI-Verbindungen)

+ lkmod

Drucker wird im Blockierbetrieb betrieben.

Backend für PostScript-Drucker

PostScript ist eine leistungsfähige Seitenbeschreibungssprache. Jedes Dokument, das Sie mit Hilfe eines postscriptfähigen Programms erzeugen, wird durch ein Programm in der Sprache PostScript beschrieben. Bei dem Ausdrucken des Dokuments arbeitet ein PostScript-Interpreter dieses Programm ab und erzeugt dabei das Bild des Dokuments auf dem Ausgabemedium. Das PostScript-Backend des SINIX-SPOOL-Systems ist für Drucker mit eingebautem PostScript-Interpreter, wie dem Siemens-Drucker 9022-300, konzipiert.

Das Backend `.../lppost` für postscriptfähige Drucker ist eine Software-Verbindung zwischen dem SINIX-SPOOL-System und dem PostScript-Interpreter des Druckers. Ohne dieses besonderes PostScript-Backend für PostScript-Drucker können Sie diese Drucker nur über das Interface-Backend betreiben. Dadurch erhalten sie in diesem Fall keine Informationen über den Betriebszustand des PostScript-Druckers. Sie können nur feststellen, ob das Interface-Backend aktiv ist. Das PostScript-Backend liefert alle Informationen über den Druckauftrag und den Betriebszustand des Druckers, die das SINIX-SPOOL-System für Benutzerinformation und Fehlerbehandlung benötigt.

Damit PostScript-Drucker und SINIX-SPOOL-System zusammenarbeiten können, müssen Sie Ihren PostScript-Drucker auf die Parameter der Datenübertragung des PostScript-Backends einstellen:

Datenflußsteuerung:

ETX/ACK

Übertragungsgeschwindigkeit:

19200 Baud

(*lpr*-Option `-speed =`)

Zeichenrahmen:

1 Startbit

8 Datenbits

Kein Paritätsbit

1 Stoppbit

(*lpr*-Option `-bits = [7|8]`)

(*lpr*-Option `-parity = [oddevenloff]`)

(*lpr*-Option `-stopb = [1|2]`)

Hinweis

Zeichenrahmen und Übertragungsgeschwindigkeit können Sie den Leistungsmerkmalen Ihres Druckers anpassen. Nach Möglichkeit sollten Sie die höchste Übertragungsgeschwindigkeit wählen, die Ihr Drucker noch fehlerfrei verarbeiten kann.

5