

ANWENDER- DOKUMENTATION	Inbetriebnahme und Wartung	MOS
3/89	Anleitung für den Systemverantwortlichen	MUTOS 1700

Programmtechnische  
Beschreibung Teil 2

Inbetriebnahme und Wartung

Anleitung für den  
Systemverantwortlichen

AC A 7100/A 7150

VEB Robotron-Projekt Dresden

Die vorliegende Dokumentation entspricht dem Stand von 3/89.

Die Ausarbeitung erfolgte durch ein Kollektiv des  
VEB Robotron-Elektronik Dresden Stammbetrieb des VEB Kombinat Robotron.

Nachdruck, jegliche Vervielfältigen oder Auszüge daraus sind unzulässig.

Herausgeber:

VEB Robotron-Projekt Dresden  
Leningrader Str. 9  
Dresden 8010

Kurzreferat

Die vorliegende Schrift ist eine Anleitung zur Inbetriebnahme und Generierung der Version 1.1 des Betriebssystems MUTOS 1700 (Multi-User-Timesharing-Operating-System) für den Arbeitsplatzcomputer A 7100/A 7150 des VEB Kombinat Robotron.

---

<b>Inhaltsverzeichnis</b>	<b>Seite</b>
1. EINLEITUNG	5
2. INBETRIEBNAHME	6
2.1. AUSLIEFERUNGSFORM DES BETRIEBSSYSTEMS	6
2.2. KOPIEREN DES AUSLIEFERUNGSSATZES	7
2.3. START DES BETRIEBSSYSTEMS	8
2.4. INSTALLIEREN DES BETRIEBSSYSTEMS AUF DER FESTPLATTE	9
2.5. HINWEISE ZUR FLOPPY-ORIENTIERTEN ARBEITSWEISE	12
2.6. INSTALLIEREN IN EINER PARTITION DER VERSION 1.0	15
3. REGENERIERUNG DES SYSTEMS	17
3.1. ÜBERBLICK	17
3.2. DIRECTORY /USR/SYS/CONF	17
3.3. DIRECTORY /USR/SYS/CFG	20
3.4. ERGEBNIS DES GENERIERUNGSLAUFES	20
3.5. ZUSÄTZLICHE SCHRITTE	20
3.6. ANSCHLUSS VON TERMINALS, DIE NICHT ZUM SYSTEMBESTAND GEHÖREN	21
Sachwortverzeichnis	22



## 1. Einleitung

~~~~~

Die Version 1.1 des Betriebssystems MUTOS 1700 ist eine Weiterentwicklung der Version 1.0, die sich vor allem dadurch auszeichnet, daß im Kern des Betriebssystems wesentliche Verbesserungen vorgenommen wurden, die eine effektivere, robustere und anwendungsfreundlichere Arbeit mit diesem System gewährleisten. Demselben Ziel dienen einige gegenüber der Version 1.0 geänderte Dienstprogramme (Kommandos). Darüber hinaus wird nunmehr der Befehlssatz des Numerikprozessors K 1810 WM 87 unterstützt.

Daraus ergeben sich folgende Ergänzungen zum Dokumentationssatz der Version 1.0 des Betriebssystems MUTOS 1700:

- Änderung der Anleitung für den Systemverantwortlichen,
- Ergänzung der Sprachbeschreibung `as` durch eine Sprachbeschreibung `asf`,
- Änderungen und Ergänzungen zum Handbuch.

Alle anderen Dokumentationen sind unverändert gültig.

Gegenüber der Anleitung für den Systemverantwortlichen der Version 1.0 gibt es wesentliche Änderungen in den Kapiteln "Inbetriebnahme" und "Regenerierung des Systems". Der neue Text ist in der vorliegenden Schrift vollständig enthalten. Das Kapitel "Wartung des Betriebssystems MUTOS" ist bis auf die unten angegebene Änderung weiterhin gültig und wurde deshalb nicht noch einmal abgedruckt. Die Änderung resultiert aus der Einteilung in die Betriebsmodi "Grundmodus" und "DCP-Modus". Der letzte Satz des Kapitels 2 auf Seite 33 ist wie folgt zu ändern:

Im Grundmodus ist über das Monitor-Kommando "b", im DCP-Modus durch Betätigen der Tastenkombination <CTRL><ALT><DEL> das Betriebssystem neu zu starten (`boot(8)`).

## 2. Inbetriebnahme

~~~~~

### 2.1. Auslieferungsform des Betriebssystems

~~~~~

MUTOS wird auf mehreren Disketten ausgeliefert, die den gesamten Umfang des Betriebssystems sowie die Hilfsmittel zum Installieren auf der Festplatte enthalten.

Ab Version 1.1 besteht die Möglichkeit, das Betriebssystem MUTOS in einer weiteren Betriebsart des Rechners zu nutzen, dem sogenannten DCP-Modus. Dieser Modus ist für A 7150 in der Grafikkonfiguration nutzbar. Im Grundmodus des Rechners arbeitet MUTOS auf dem A 7100 und auf dem A 7150 mit alphanumerischer Konfiguration. Diese beiden unterschiedlichen Betriebsarten des Rechners schlagen sich in unterschiedlichen Generierungen des MUTOS und in unterschiedlich aufgebauten BOOT-Disketten nieder.

Der Auslieferungssatz umfaßt 11 Disketten. Für jede Betriebsart gibt es eine spezifische BOOT-Diskette (ROOT1). Die restlichen 9 Disketten gelten für beide Varianten.

Nach Möglichkeit sollte MUTOS im DCP-Modus betrieben werden. Der Leistungsumfang in dieser Betriebsart ist größer. Außerdem ist die Übereinstimmung mit dem Betriebssystem MUTOS 1834 für den ESER-PC EC 1834 sehr hoch, was für die Portabilität von Programmen interessant ist.

Die BOOT-Disketten haben folgenden Aufbau:

```
Grundmodus:  Spur      0          - FM, 16*128 Byte
              Spuren    1...159    - MFM, 16*256 Byte
```

```
    ROOT-File-System-Format "dxmf" mit 636 kByte
    (Spuren 1...159)
```

```
DCP-Modus:   Spuren    0...159    - MFM, 9*512 Byte
```

```
    ROOT-File-System-Format "dmf" mit 720 kByte
```

Die Spur 0 der BOOT-Diskette im Grundmodus und die Sektoren 0 und 1 der Spur 0 im DCP-Modus enthalten jeweils ein MUTOS-spezifisches Boot-Programm. Dieses kann durch das Monitor-Kommando `boot(b)` aktiviert werden und lädt das File `/mutos` aus dem File-System in den Hauptspeicher (Systemstart). Das Root-File-System enthält einen Basisumfang an Komponenten, der für das Installieren auf der Festplatte benötigt wird.

Alle anderen Disketten des Liefersatzes sind im Format

```
    DS, DD, 80*9*512 Byte ("dmf"-Format).
```

Sie enthalten im MUTOS-tar-Format alle restlichen Komponenten des Betriebssystems, die sich nicht auf der ROOT1-Diskette befinden.

Alle Disketten werden schreibgeschützt ausgeliefert. Ab Version 1.1 ist auch das File-System auf der BOOT-Diskette in diesem Zustand. Deshalb sind Arbeiten mit der BOOT/ROOT-Diskette nur möglich, wenn keine Schreibzugriffe erfolgen. Für einen Nur-Disketten-Betrieb wird eine sinnvolle Herangehensweise im Abschnitt 2.5. erläutert.

## 2.2. Kopieren des Auslieferungssatzes

Sollen die Disketten vor der aktiven Nutzung gedoppelt werden, geschieht das am besten physisch unter einem anderen Betriebssystem. Als Beispiel wird das Kopieren unter SCP mittels COPYDISK beschrieben (Eingaben sind mit <CR> abzuschließen):

| Ausschrift von COPYDISK                                                                      | Eingabe beim Kopieren     |                             |
|----------------------------------------------------------------------------------------------|---------------------------|-----------------------------|
|                                                                                              | von ROOT1<br>(Grundmodus) | der restlichen<br>Disketten |
| Source disk?                                                                                 | A                         | A                           |
| Using current drive parameters<br>(N/<CR>)?                                                  | N                         | N                           |
| Cyls/Disk :80                                                                                | <CR>                      | <CR>                        |
| Tracks/Cyl :2                                                                                | <CR>                      | <CR>                        |
| Sectors/Track :16                                                                            | <CR>                      | 9                           |
| Bytes/Sector :256                                                                            | <CR>                      | 512                         |
| Track 0 handling? Yes                                                                        | <CR>                      | N                           |
| Destination disk?                                                                            | B                         | B                           |
| Format Destination Disk<br>(Y/N)?                                                            | Y                         | Y                           |
| Interleave :1                                                                                | 4                         | 4                           |
| Insert destination disk in drive B<br>Type <CR> to continue                                  | <CR>                      | <CR>                        |
| Insert source disk in drive A<br>Insert destination disk in drive B<br>Type <CR> to continue | <CR>                      | <CR>                        |

Unter DCP sind die Programme DISKCOPY bzw. MDISK zum physischen Kopieren geeignet.

Unter MUTOS wird folgende Herangehensweise empfohlen:

Root-Diskette (Grundmodus):

Mittels `mkb(8)` wird auf einer leeren Diskette das Formatieren und das Schreiben des Boot-Programms auf die Spur 0 vorgenommen.  
Das Root-File-System wird mit `dd(1)` physisch übertragen (spurweises Kopieren):

```
dd if=/dev/rdxmf0 of=/dev/rdxmf1 bs=4k
```

Tar-Disketten und Root-Diskette (DCP-Modus):

Formatieren der Disketten:

```
format /dev/rdmf?
```

Physisches Übertragen (2 Spuren bei jedem Übertragungsschritt):

```
dd if=/dev/rdmf0 of=/dev/rdmf1 bs=9k
```

Sollen mehrere Kopien erstellt werden, kann auf einem System mit Festplatte auch folgender Weg beschritten werden:

übertragen des gesamten Disketteninhalts als File auf die Festplatte:

```
dd if=/dev/rdxmf? of=ROOT1 bs=4k (Root-Diskette (Grundmodus))
dd if=/dev/rdmf? of=ROOT2 bs=9k (Bsp.für tar-Disketten)
```

Vorbereiten der Zieldisketten über `mkb(8)` bzw. `format(8)` wie oben beschrieben. Übertragen der Files auf die neuen Disketten:

```
dd if=ROOT1 of=/dev/rdxmf? bs=4k (Root-Diskette (Grundmodus))
dd if=ROOT2 of=/dev/rdmf? bs=9k (Bsp. für tar-Disketten)
```

Dieser Weg ist zeitlich effektiver, besonders beim Herstellen mehrerer Kopien.

### 2.3. Start des Betriebssystems

Nach dem Einschalten des Rechners bzw. nach Betätigen der RESET-Taste läuft das "Confidence"-Testprogramm ab (Firmware des Rechners). Die zu dem vorgesehenen Modus gehörende Diskette ROOT1 ist in das Laufwerk 0 einzulegen und dieses bereit zu machen. Nach erfolgreichem Hardware-Test bzw. durch das Monitor-Kommando "b<CR>" wird der Betriebssystemkern eingelesen und gestartet (siehe auch `boot(8)`). Die vorgesehenen Anschlußsteuerungen werden auf ihr Vorhandensein getestet, wobei entsprechende Ausschriften erscheinen.

Im Grundmodus ist auch ein Booten vom Laufwerk 1 möglich. Deshalb werden an dieser Stelle Meldungen über die generierten Werte für RAM-Floppy-Emulation, Root-, Pipe- und Swap-Gerät sowie für Beginn und Größe des Swap-Bereiches angezeigt. Diese Standardwerte können verändert werden. Es ist nicht erforderlich, wenn die Root-Diskette im Laufwerk 0 liegt. Dann kann die Abfrage

Do you want changes?

verneint werden (<CR>).

Danach erscheint eine Betriebssystemmeldung mit Angabe einer Versionsnummer und dem für Nutzerprozesse zur Verfügung stehenden Hauptspeicherbereich (in kByte).  
Nach

single user login:

meldet man sich mit `root<CR>` an. Daraufhin erscheint das Zeichen "#" vom Kommandoprozessor `shell (sh(1))` als Kennzeichen der Eingabebereitschaft. Jede eingegebene Zeile ist mit <CR> oder <LF> abzuschließen. Vor Abschluß der Eingabezeile ist eine Zeichenirrtung über die Taste <DEL>, eine Zeilenirrtung über CTRL-U erreichbar.

In den nächsten beiden Abschnitten wird die unterschiedliche Herangehensweise beim Installieren des Betriebssystems auf der Festplatte und beim Vorbereiten einer Floppy-orientierten Arbeit erläutert.

## 2.4. Installieren des Betriebssystems auf der Festplatte

~~~~~

Das Installieren des Betriebssystems MUTOS kann unabhängig von der Vorgeschichte der Festplatte und von ihren Parametern erfolgen. Sind bereits andere Betriebssysteme auf der Platte eingerichtet, kann MUTOS in einem zusammenhängenden freien Bereich von mindestens 6 MByte untergebracht werden. Bei kleineren Bereichen ist höchstens eine Teilinstallation möglich. Die anderen Systeme werden dabei nicht beeinflusst. Bei unformatierten oder noch nicht eingerichteten Platten übernimmt die Installationsprozedur das Formatieren und/oder das Einrichten von Partitions. Dabei wird eine vorhandene Defektspurliste berücksichtigt. Ist keine auf der Platte eingetragen, können defekte Spuren von Hand eingegeben werden. Sie werden als Defektspurliste auf die Platte geschrieben und beim Formatieren speziell behandelt.

Die zum Auslieferungsumfang gehörende ROOT1-Diskette enthält Shell-Prozeduren und Programme für folgende Funktionen:

- Modifikation bzw. Errichtung eines Master-Boot-Blocks (Zylinder 0, Spur 0, Sektor 0).
- Einrichten einer MUTOS-Partition.
- Einrichten der Subpartitions für Root-, User-, Temp- und Swap-Bereich.
- Formatieren der MUTOS-Partition in geeigneter Form.
- Installieren des MUTOS-Boot-Programms auf der 1. Spur der MUTOS-Partition.
- Einrichten der File-Systeme für Root-, User- und Temp-Bereich.
- Übertragen aller Files von den Disketten auf die Platte.

Das Einrichten einer DCP-Partition mit der Installationsprozedur ist nicht möglich.

Nach dem Start des Betriebssystems von der Diskette entsprechend Abschnitt 1.3. wird das Installieren auf der Festplatte über die Shell-Prozedur `mkhd(8)` eingeleitet:

`mkhd`

Innerhalb dieser Prozedur wird zunächst das Programm `fdisk(8)` und später das Programm `hdinstall(8)` abgearbeitet. Es wird ausgetestet, ob auf der Festplatte

ein Master-Boot-Block vorhanden und eine MUTOS-Partition eingerichtet ist. Die Parameter der Festplatte und die Partition-Aufteilung werden angezeigt. Zuerst wird die bisher gültige Liste defekter Spuren zur Modifikation angeboten. Danach kann die aktuelle Partition-Tabelle modifiziert werden, wobei die gewünschte Partition vom Typ MUTOS und aktiv (Kennzeichen A) sein muß. Im nächsten Menue wird die logische Aufteilung innerhalb der MUTOS-Partition zur Modifikation bereitgestellt.

```
Standard:
          Root : 20%           Root-File-System (/)
          Swap : 10%          Auslagerungsbereich
          Temp : 10%          Temp-File-System (/tmp)
          User : 60%          User-File-System (/usr)
```

Die Steuerung der angebotenen Menues erfolgt über folgende Tasten:

```
<CR>      Übergang zum Eingabemodus (Auswahl der blinkenden Funktion)
HOME      Übergang ins Hauptmenue
->
<-        \
           | Positionierung innerhalb
           | von Eingabefeldern
           | bzw.
           | Menue-Srukturen
           /
A/N       aktiv bzw. passiv setzen in der Statusspalte
M/m      Kennzeichen für MUTOS in der Systemspalte
DEL      Löschen des Partition-Eintrages
```

In den Spalten "Beginn" und "Ende" wird der Cursor automatisch auf die vorderste Stelle des bisherigen Wertes positioniert. Mit den Cursor-Funktionstasten kann auf die zu modifizierende Stelle übergegangen werden, auch auf bisher unbelegte Plätze (z.B. beim Ausgangswert 0).

Danach wird abgefragt, ob ein Formatieren des MUTOS-Bereiches auf der Platte vorgenommen werden soll. Bei einer Erstinstallation oder bei veränderter Defektpurlliste im Bereich der MUTOS-Partition ist das Formatieren vorzunehmen.

Das Programm `fdisk` zeigt während des Formatierungslaufs die aktuellen Zylinder/Kopf-Wertepaare absolut zum Plattenbeginn an. Außerdem wird sichtbar gemacht, wenn vorgegebene defekte Spuren als solche formatiert werden. Zusätzlich erkannte defekte Spuren werden gemeldet und in die Defektpurlliste aufgenommen.

Es erfolgt nun eine interne Überprüfung des Swap-Bereiches. Beinhaltet dieser Bereich defekte Spuren, so muß die Lage dieser Partition geändert werden. Danach werden zwei Modi unterschieden:

- a.) Installation des Systems
- b.) Reparatur des Systemkerns auf dem Root-Filesystem

Im Falle der Installation wird der Boot-Block kopiert und die File-Systeme für `root`, `user` und `temp` eingerichtet. Abfragen ermöglichen es, dies schrittweise zu übergehen. Bei der Erstinstallation müssen diese Schritte alle ausgeführt werden. Beim Einrichten der File-Systeme über `mkfs(1)` werden Blöcke, die auf defekten Spuren liegen, durch Eintragen in die Inode 1 von der späteren Verwendung ausgeschlossen. Dieser Vorgang nimmt einige Zeit in Anspruch, da eine Schreib- und Leseprüfung erfolgt.

Als nächster Schritt wird das Root-File-System der Platte an `/mnt` angebunden. Dann wird das File `mutos` von der Diskette auf die Platte übertragen und dort den aktuellen Platten- und Partition-Parametern angepaßt. Zum Abschluß wird noch ein Text-File `diskpar` in die Root-Directory auf der Platte geschrieben. Es enthält die Platten- und Partition-Parameter, wie sie für die Generierung in `/usr/sys/cfg/c5170.c` benötigt werden. Die Zelle "Genboot" im File `mutos` wird rückgesetzt und die Ausgabe des Anfangsdialogs beim Systemstart dadurch unterdrückt. Der weitere Ablauf wird wieder durch `mkhd(8)` gesteuert. Der Inhalt der Diskette `ROOT1` wird auf die Platte übertragen, die Geräteeinträge in `/dev` werden aufgebaut und einige zusätzliche Schritte ausgeführt. Zum Schluß der Installationsprozedur ist ein arbeitsfähiges MUTOS auf der Platte vorhanden, das unmittelbar gestartet werden kann. Der gesamte Vorgang erfolgt dialoggesteuert bzw. automatisch.

Die Prozedur `mkhd(8)` zeigt zum Abschluß an, wie ein Systemstart von der Platte vorgenommen werden kann. Danach geht das bis zu diesem Zeitpunkt laufende MUTOS von der Vertriebsdiskette in einen definierten Systemhalt.

Die Systemvarianten "Grundmodus" und "DCP-Modus" unterscheiden sich im Systemkennzeichen der Partition im Master-Boot-Block. Das neue Kennzeichen für den DCP-Modus wird vom Monitor des Rechners ab Version 2.4 unterstützt. Ältere Versionen bringen beim Kommando "a" für eine MUTOS-Partition im DCP-Modus "???" anstelle "MUTOS". Dies ist nur eine Frage der Anzeige und wirkt sich auf den Boot-Prozeß und die weitere Arbeit nicht aus.

Nach Neustart des Systems von der Platte ist die Shell-Prozedur `install(8)` zu starten, die den Inhalt der restlichen Vertriebsdisketten auf die Platte überträgt. Auch dies geschieht dialoggesteuert.

Sollte eine der Disketten Übertragungsfehler anzeigen, ist eine der lesbaren tar-Disketten zweimal einzulegen. Natürlich fehlen dann Komponenten, doch kann so die Prozedur `install` fortgesetzt werden. Nach Ersatz der defekten Diskette kann jederzeit nachträglich eine Übertragung des Inhaltes mit `tar(1)` erfolgen.

Für eine Neugenerierung sind die Werte aus `/diskpar` nach `/usr/sys/cfg/c5170.c` zu übernehmen. Zu beachten ist noch, daß nach dem Neustart des Systems von der Festplatte die File-Systeme

```
und          /dev/user an /usr
             /dev/temp an /tmp
```

zu binden sind (`mount(1)`).

Im Falle einer Reparatur des bereits installierten Systems werden folgende Schritte ausgeführt:

- File-System-überprüfung für Root-File-System (dialoggesteuert)
- Angliedern des Root-File-Systems (`mount`)
- Übertragung des Systemkerns
- Ausgliedern des Root-File-Systems (`umount`)

Damit ist ein lauffähiger Kern neu installiert und es können eventuelle Fehler beseitigt werden.

Typische Fehlerquellen sind:

- Boot-Block zerstört (**ibblock**(8) benutzen)
- **/etc/init** bzw. **/etc/inir** defekt
- Eintrag **/dev/console** defekt
- **/bin/sh** defekt

Die defekten Komponenten können von der Root-Diskette neu installiert werden.

Zwei Schritte sollten zum Abschluß der Installation noch ausgeführt werden:

- Das File **diskpar** wird für eventuelle spätere Reparaturen mit

```
cat /diskpar > /dev/lp
```

ausgedruckt und die Liste archiviert.

- Es wird eine physische Kopie der Diskette ROOT1 des Vertriebssatzes hergestellt (siehe Abschnitt 2.2.) und das auf der Platte mit angepaßten Parametern vorhandene neu installierte File **mutos** auf die Kopie überspielt.

```
/etc/mount /dev/dmf0 /mnt      (Kopie im Laufwerk 0)
```

```
cp /mutos /mnt                (Kopieren von der Platte zur Diskette)
```

Danach sind im File **/mnt/mutos** die folgenden Parameter mittels **adb**(1) auf die gleichen Werte einzustellen, wie sie das Auslieferungssystem hat:

```
rootdev, pipedev, swapdev
swplo, nswap
Bigswap
```

Das Beispiel am Ende des Abschnittes 2.5. kann als Anleitung für die Arbeit mit **adb** dienen.

Dieses Retten des aktuellen Systems, das die richtigen Plattenparameter enthält, ist auch nach einer Neugenerierung zu empfehlen. Sollte sich das Plattensystem einmal nicht starten lassen, kann damit eine Reparatur des File-Systems von der Diskette aus vorgenommen werden, ohne **mkhd** abarbeiten zu müssen.

## 2.5. Hinweise zur Floppy-orientierten Arbeitsweise

Für diese Arbeitsweise ist es erforderlich, daß der Nutzer sein System so zusammenstellt, wie es seinem Aufgabenspektrum entspricht. Eine allgemeingültige Variante kann es aufgrund der begrenzten Externspeicherkapazität nicht geben.

Für diese Arbeiten sind gute Kenntnisse des MUTOS erforderlich. Sie sollten erfahrenen Systemprogrammierern vorbehalten bleiben.

Folgende Hinweise können gegeben werden:

1. Da die Diskette ROOT1 logisch und physisch schreibgeschützt ist, kann sie für das Erstellen neuer Dateien nicht unmittelbar genutzt werden. Es ist deshalb erforderlich, zunächst eine physische Kopie dieser Diskette herzustellen (siehe 1.2.). Diese neue Diskette wird über **mount**(1) an die Directory **/mnt** der ROOT1-Diskette angegliedert. Mittels **adb**(1) kann nun das File **mutos** auf der neuen Diskette modifiziert werden (siehe auch Beispiel am Ende dieses Abschnitts):

Die Zelle R0root1 (bisher auf 1 gesetzt) ist mit 0 zu füllen. Die Anzeige der generierten Werte für `rootdev`, `pipede` usw. nach dem Booten des Betriebssystems wird über die Zelle `Genboot` gesteuert und kann nach Wunsch eingestellt werden (0 : keine Anzeige/Änderungsmöglichkeit, 1 : Anzeige/Änderungsmöglichkeit).

Diese so veränderte Boot/Root-Diskette kann nun anstelle der ROOT1-Diskette auch für Schreibzugriffe genutzt werden.

2. Löschen aller Komponenten von der Root-Diskette, die für die Installation auf der Festplatte erforderlich sind (`mkhd`, `install`, `hdinstall`, `fdisk`).
3. Aufbau einer getrennten Boot-Diskette, damit das File `/mutos` nicht auf der Root-Diskette geführt werden muß (`mkbf(8)`). Dabei sollte die Zelle `Genboot` gesetzt werden, um zum Zeitpunkt der Abfrage einen Diskettenwechsel vornehmen zu können (Austausch Boot-Diskette gegen Root-Diskette).
4. Das Root-File-System ist im Format "dmf" oder "bdmf" aufzubauen, um die Diskette maximal auszunutzen.
5. Im Hauptspeicher kann ein Teil als Swap-Bereich reserviert werden. Seine Größe sollte etwa 20% des für Nutzerprozesse insgesamt zur Verfügung stehenden Bereiches betragen.
6. Es werden "User"-Disketten aufgebaut, die an `/usr` angebunden werden können (`mount(1)`) und aufgabenspezifisch zusätzliche Systemkommandos sowie Nutzerkomponenten enthalten.  
Beim Kopieren der Standardbibliothek `/lib/libc.a` ist so vorzugehen, daß die Bibliothek auf ein File-System mit möglichst viel freiem Platz kopiert und gleich danach `ranlib(1)` für das neue File gestartet wird.
7. Für bestimmte Zwecke ist es günstig, einen weiteren RAM-Bereich für das Einrichten eines File-Systems zu nutzen. Mit den RAM-Bereichen kann wie mit Externspeichern gearbeitet werden (`mkfs(1)`, `mount(1)`, `umount(1)`). Über diese RAM-File-Systeme können Kopiervorgänge vorgenommen werden, Programme für schnelles Abarbeiten sind dort stationierbar, und als Bereich für temporäre Files erhöht diese Arbeitsweise die Abarbeitungsgeschwindigkeit.
8. Beim Vorhandensein von nur zwei Diskettenlaufwerken ist es nicht möglich, den gesamten Inhalt einer tar-Vertriebsdiskette in einem Arbeitsgang einzulesen. In Teilen kann das Einlesen über einen RAM-Floppy-Bereich oder über die Root-Diskette erfolgen, wenn dort genügend Platz vorhanden ist. Am einfachsten ist es jedoch, diese Vorbereitungen für die Floppy-orientierte Arbeit auf einem Plattensystem zu machen.
9. Eine Reihe von Systemparametern kann ohne Neugenerierung geändert werden. Einige Möglichkeiten sind im folgenden Beispiel aufgezeigt.

Beispiel (Bedienereingaben sind hervorgehoben):

```
adb -w mutos           Aufruf adb mit Schreiberlaubnis
~~~~~

Genboot?*d           Abfrage "Genboot" dezimal
~~~~~

_Genboot:    1           Ausgabe durch adb
?*w 0
~~~~~

_Genboot:    01 = 0      Ausgabe der Änderung durch adb
R0root?*d         Abfrage "R0root" dezimal
~~~~~

_R0root:     1           Ausgabe durch adb
?*w 0
~~~~~

_R0root:     01 = 0      Ausgabe der Änderung durch adb
swapdev?*d        Abfrage "swapdev" dezimal
~~~~~

_swapdev:    44          Ausgabe durch adb

?*w 256
~~~~~
_swapdev:    054 = 0400  Major/Minor-Nr. von /dev/ram0
                        eingeben
                        Ausgabe der Änderung durch adb

swplo?*D          Abfrage "swplo" long-dezimal
~~~~~

_swaplo:     1800        Ausgabe durch adb

?*W 0
~~~~~
_swaplo:     03410 = 0    Neuer Wert von "swplo"
                        Ausgabe der Änderung durch adb

nswap?*d          Abfrage "nswap" dezimal
~~~~~

_nswap:      900         Ausgabe durch adb

?*w 112
~~~~~
_nswap:      01604 = 0160 Neuer Wert von "nswap"
                        Ausgabe der Änderung durch adb

ramcfg,8?*d       Abfrage der Tabelle "ramcfg"
~~~~~

_ramcfg:      0    0    0    0    0    0    0    0
```

---

```

                                     Ausgabe von adb

ramcfg,2?*w 400 112      Ändern des Wertpaares für /dev/ram0
~~~~~
_ramcfg:    0 = 0620      Ausgabe der Änderung durch adb
_ramcfg+#2: 0 = 0160

$q                                     Verlassen des adb
~~~~

```

Beim Boot-Prozeß des so modifizierten Systems wird `/dev/ram0` als Swap-Gerät ausgewiesen. Der Swap-Bereich beginnt ab 0 innerhalb des Bereiches von 400 KByte bis 512 KByte und belegt diesen vollständig.

ähnliche Änderungen sind für andere Parameter ebenfalls möglich, wenn sie Hauptspeicherplatz belegen. Die Header mit den entsprechenden Definitionen sind in `/usr/include/sys` enthalten.

Die Zelle "Genboot" unterdrückt beim Wert 0 den Anfangsdialog. Das ist in dem Fall günstig, wenn die betreffenden Parameter der Konfiguration angepaßt sind und nicht verändert werden sollen.

Als weiteres Beispiel für eine derartige Änderung soll die Einstellung der Schrittrate für die verschiedenen Floppy-Laufwerktypen genannt werden. Dafür ist das File `/usr/sys/cfg/c5170.c` im Zusammenhang mit dem Header `/usr/sys/h/k5170.h` als Orientierung zu benutzen (siehe auch [kes\(4\)](#)).

## 2.6. Installieren in einer Partition der Version 1.0

Prinzipiell ist es möglich, eine schon vorhandene Installation der Version 1.0 zu erhalten und zur Version 1.1 aufzuwerten. Dieser Schritt sollte aber die Ausnahme bleiben, da sich die Einteilung der Subpartitions und die Einrichtung der File-Systeme verändert haben. Die Effektivitätserhöhung der Version 1.1 würde dabei unwirksam bleiben. Außerdem ist dieser Übergang nicht ganz unproblematisch und sollte, wenn unbedingt erforderlich, von erfahrenen Systemverantwortlichen vorgenommen werden. Günstiger ist es in jedem Fall, nutzerspezifische Daten zu retten und eine Neuinstallation vorzunehmen.

Besteht jedoch die Notwendigkeit dieses Übergangs, sind folgende Schritte auszuführen:

- Das File `mutos` des bisherigen Systems auf der Platte ist in `mutos.old` umzubenennen:

```
mv /mutos /mutos.old
```

(Der Name `mutos.old` wird zum Entnehmen der Partition-Parameter des bisherigen Systems benötigt.)

- Das System der neuen ROOT1-Diskette ist anzubooten und `mkhd` zu starten (siehe Abschnitt 2.3. und 2.4.).
  - Die MUTOS-Partition ist aktiv zu setzen
  - Das Defektspurlisten-Menue ist zu übergehen.
  - Das Subpartition-Menue ist zu übergehen.
  - Das Formatieren ist zu übergehen.
  - Die Abfrage "Vollständige Installation ?:" ist zu verneinen (Eingabe: **n**).
  - Die Abfrage "Installation auf bereits vorhandenes System ?:" ist mit **y** zu beantworten.
  - Nach Abschluß der `mkhd`-Prozedur wird das neue System wie bei einer Neuinstallation gestartet und über `install` vervollständigt. Dabei werden die  
MUTOS 1700
- 

alten Komponenten des Betriebssystems durch die neuen überschreibt. Die Nutzer-Files auf allen File-Systemen bleiben erhalten, wenn keine Namensübereinstimmungen mit System-Files bestehen.

Nutzerspezifische Einträge in der `/dev`-Directory gehen bei dieser Art der Installation verloren und müssen wieder neu eingerichtet werden.

### 3. Regenerierung des Systems

~~~~~

#### 3.1. Überblick

~~~~~

MUTOS wird in Form eines Basissystems ausgeliefert, das nicht unbedingt den konkreten Anforderungen des Anwenders entsprechen wird. Deshalb besteht die Möglichkeit, durch Generierung eines neuen Systemkerns mit veränderten Parametern und Einbeziehung anderer Geräte eine einsatzfallspezifische Anpassung vorzunehmen. In diesem Kapitel wird dafür eine Anleitung gegeben. Die Generierung geschieht aus Quellmodulen. Alle benötigten Module und Kommando-Files sind im Vertriebssystem in sechs Subdirectories von `/usr/sys` enthalten:

<code>/usr/sys/h</code>	Header-Files, die über "#include..." in Systemmodulen benutzt werden
<code>/usr/sys/io</code>	Geräte-Driver und einige angelehnte Module
<code>/usr/sys/sys</code>	überwiegender Teil des Systems
<code>/usr/sys/mdep</code>	Maschinenabhängige Module
<code>/usr/sys/cfg</code>	Files, die Adressen und Tabellen enthalten
<code>/usr/sys/conf</code>	Allgemeine Files zur Generierung

Diese sechs Subdirectories können auch an beliebiger anderer Stelle im File-System angeordnet werden. Sie bilden einen in sich abgeschlossenen Komplex. Für den Generierungslauf wird das Systemprogramm `make(1)` genutzt. Das benötigte `Makefile` steht in `/usr/sys/conf`. Nach Beendigung von `make` steht der neue Systemkern als File mit dem Namen `mutos` in `/usr/sys/conf`. Wesentliche Eigenschaften des Systems werden in `/usr/sys/h/param.h` festgelegt. Daran sollte nichts verändert werden. Andere Parameter werden im `Makefile` bzw. in `c.c` in `/usr/sys/conf` eingestellt. Darauf wird im Punkt 3.2. eingegangen.

#### 3.2. Directory `/usr/sys/conf`

~~~~~

Diese Directory enthält das File `Makefile`, das als Eingabe-File für `make(1)` dient. Es ist so aufgebaut, daß es ohne Editiervorgänge den gewünschten Parametern angepaßt werden kann. Diese Parameter betreffen optionale Driver, den gewünschten Modus (Grund- oder DCP-Modus) und zusätzliche Kernteile. Das Einbinden bzw. Weglassen der optionalen Teile und die Moduseinstellung werden über `make`-Aufrufe der Form

```
make XXX [YYY] ... bzw.
make noXXX [noYYY] ...
```

vorbereitet. Setzen und Rücksetzen sind auf einer Zeile gemischt möglich. Die Anzeige der eingestellten Parameter kann durch

```
make show
```

erreicht werden. Der `make`-Lauf für Veränderungen von Parametern nimmt einige Zeit in Anspruch, da umfangreiche Suchvorgänge nach den betroffenen Modulen erfolgen.

Für "XXX" können folgende Werte stehen:

|         |   |                                                                       |
|---------|---|-----------------------------------------------------------------------|
| LP      | - | Drucker-Driver                                                        |
| IFSS    | - | Driver für Interface IFSS (über ASP)                                  |
| V24     | - | Driver für Interface V24 (über ASP)                                   |
| IFSP    | - | Driver für Parallelinterface (über ASP)                               |
| AMX     | - | Driver für Controller P516                                            |
| RAM     | - | Driver für RAM-Floppy-Emulation                                       |
| 7100    | - | Grundmodus                                                            |
| 7150    | - | DCP-Modus                                                             |
| MSDOS   | - | Abarbeitung von Programmen der Cross-Entwicklungsumgebung ermöglichen |
| LOAD86  | - | Start von Standalone-Programmen im Objekt-Modul-Format ermöglichen    |
| KDEBUG  | - | Testhilfe für Kern-Testung einbinden                                  |
| SYSACCT | - | Unterstützung des Systemrufes <code>sysacct(2)</code>                 |
| SYSLOCK | - | Unterstützung des Systemrufes <code>syslock(2)</code>                 |
| DISKMON | - | Führen von Statistikinformationen zu Externspeichern                  |
| INSTRM  | - | Führen weiterer Statistikinformationen                                |

Es ist nicht möglich, alle Optionen gleichzeitig einzubinden. Der Code des Kerns überschreitet in diesem Fall den erlaubten Wert von 64kByte. Die Optionen `MSDOS`, `LOAD86`, `KDEBUG`, `SYSLOCK`, `DISKMON` und `INSTRM` sollten speziellen Lösungen vorbehalten bleiben.

Achtung!

Wird der Modus des Betriebssystems geändert, und das neu generierte System soll anstelle des bisher im anderen Modus laufenden Kerns gestartet werden, sind zusätzliche Schritte erforderlich (siehe Abschnitt 3.5.).

Weitere Veränderungen sind in `c.c` möglich. über Parameter werden die Größen verschiedener systeminterner Tabellen sowie Grenzwerte festgelegt. Die ordnungsgemäße Funktion des Gesamtsystems hängt von einer geeigneten Wahl dieser Parameter ab. Dieser Abschnitt soll dafür Erläuterungen geben. Im allgemeinen ist eine Änderung der folgenden Werte nicht notwendig.

`NBUF` Das System MUTOS arbeitet mit einem Cache-Puffer-Mechanismus für Plattenspeicher. Je größer die Anzahl dieser Puffer ist, um so effektiver erfolgt die Plattenarbeit.

---

|         |                                                                                                                                                                                                                                                                                                                                                                             |
|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| NINODE  | Legt die Größe der Inode-Tabelle fest. Jeweils einen Eintrag in dieser Tabelle benötigt <ul style="list-style-type: none"> <li>- jedes geöffnete File</li> <li>- jede Arbeits-Directory</li> <li>- jedes "sticky"-Text-Segment</li> <li>- jedes geöffnete Gerät</li> <li>- jedes über <code>mount</code> eingegliederte Gerät</li> </ul> Pro Inode werden 74 Byte benötigt. |
| NFILE   | Legt die maximale Anzahl geöffneter Files fest. Bei jedem <code>open(2)</code> oder <code>creat(2)</code> für ein File wird eine Eintragung vorgenommen. Jede Eintragung belegt 8 Byte.                                                                                                                                                                                     |
| NMOUNT  | Legt die maximale Anzahl eingegliedelter File-Systeme fest. Das Root-File-System ist zu berücksichtigen.                                                                                                                                                                                                                                                                    |
| MAXUPRC | Maximale Anzahl von Prozessen, die ein Nutzer gleichzeitig aktiviert haben darf. Der Wert sollte so festgelegt werden, daß sinnvoll gearbeitet werden kann, jedoch ein Nutzer nicht alle möglichen Prozesse belegt.                                                                                                                                                         |
| MAPSIZ  | Bestimmt die Anzahl von Fragmenten, in die der Hauptspeicher bzw. der Auslagerungsbereich unterteilt werden können. Das theoretische Maximum liegt bei der doppelten Anzahl von möglichen Prozessen (2 x NPROC). Es werden 8 Byte x (MAPSIZ + 1) belegt.                                                                                                                    |
| NCALL   | Bestimmt die Anzahl möglicher simultaner Aufrufe der Timeout-Routine (systemintern). Die Callout-Tabelle belegt 6 Byte/Aufruf.                                                                                                                                                                                                                                              |
| NPROC   | Legt die maximale Zahl von Prozessen fest und hängt vom Anforderungsspektrum typischer Nutzer ab. Je Prozeß werden 32 Byte benötigt.                                                                                                                                                                                                                                        |
| NTEXT   | Bestimmt die maximale Anzahl reiner Textsegmente (Code), d.h. simultan arbeitender reiner Programme. Pro Textsegment werden 11 Byte belegt.                                                                                                                                                                                                                                 |
| NCLIST  | Anzahl von Segmenten der "clist", in der die Daten zeichenweise arbeitender Geräte zwischengespeichert werden. Jedes Segment enthält Speicherplatz für 6 Zeichen. NCLIST sollte so groß gewählt werden, daß pro Terminal etwa 80 Zeichen (eine Zeile) zwischengespeichert werden können.                                                                                    |

Entsprechend den Angaben zu den einzelnen Parametern können solche Werte gewählt werden, die ein effektives System entstehen lassen. Als Orientierung ist `c.c` in der Auslieferungsform zu nutzen. Die dort angegebenen Parameter wurden für das ausgelieferte Basissystem eingestellt. Die Größe bestimmter Komponenten (z.B. Tabellen, Felder, Routinen, Driver) ist aus der Symboltabelle des Files `/mutos` ersichtlich, die man mit `nm(1)` erhält. Die Werte für die beim Systemstart eingestellten Größen `rootdev`, `pipeDEV`, `swapdev`, `swplo` und `nswap` sind ebenfalls in `c.c` zu finden und bei Bedarf zu modifizieren. Dafür wird ein Makro `makeDEV` genutzt, dessen Parameter die Major- und Minor-Nummer entsprechend den Einträgen in `/dev` sind.

Die Parameter für GENBOOT , ROROOT und BIGSWAP sind nach folgenden Gesichtspunkten zu wählen:

|         |                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| GENBOOT | Ein Wert ungleich 0 bewirkt, daß beim Systemstart die generierten Werte für <code>rootdev</code> , <code>pipeDEV</code> , <code>swapdev</code> , <code>swplo</code> , <code>nswap</code> und RAM-Floppy-Emulation angezeigt werden. An dieser Stelle sind Änderungen dieser Werte möglich.<br>Ein Wert 0 für GENBOOT übergeht diesen Dialog. Diese Variante ist günstig, wenn alle oben genannten Parameter den richtigen Wert besitzen. |
| ROROOT  | Der Parameter ist nur für Floppy-Arbeit ungleich 0 zu setzen. Dies ist für die Diskette ROOT1 des Auslieferungssatzes der Fall und kann auch bei speziellen Anwendungen genutzt werden.                                                                                                                                                                                                                                                  |

BIGSWAP Über diesen Parameter wird der Swapping-Algorithmus beeinflusst. Liegt der Swap-Bereich (swapdev) auf einer Platten-Partition, ist BIGSWAP auf 1 zu setzen. Bei Floppy-Arbeit ist BIGSWAP = 0 erforderlich.

### 3.3. Directory /usr/sys/cfg

Diese Directory enthält Files für die einzelnen E/A-Schnittstellen zur Festlegung von Adressen und bestimmten Parametern. Änderungen entsprechend der gewünschten Konfiguration sind hier möglich. Von Bedeutung sind vor allem die Files `c5170.c` für Platte und Floppy und `cram.c` für eventuell gewünschte Standardeinstellungen zur Behandlung von Hauptspeicherbereichen in spezieller Form. In `c5170.c` können die konkreten Parameter der Festplatte (Anzahl Zylinder/Spuren/Sektoren, Partition-Aufteilung) modifiziert werden (siehe auch `kes(4)`). Prinzipielle Änderungen an den Tabellen sind zu vermeiden (z.B. Streichen von Einträgen), da dadurch die Bezüge zu den Minor-Nummern in `/dev` verändert werden.

### 3.4. Ergebnis des Generierungslaufs

Sind alle gewünschten Parameter und Optionen eingestellt, wird über

```
make mutos
```

die eigentliche Generierung angestoßen.

In der Directory `/usr/sys/conf` entsteht das File `mutos`. Es kann für eine Floppy-orientierte Arbeitsweise über `strip(1)` noch um die Symboltabelle verkürzt werden.

Für die Erprobung des neuen Systems ist es günstig, das bisher benutzte `/mutos` aufzuheben, z.B. durch Umbenennung:

```
mv /mutos /mutos.old
```

Danach kann das neue System in die Root-Directory kopiert werden:

```
cp mutos /mutos
```

Arbeitet das neue System nach `boot(8)` ordnungsgemäß, kann `/mutos.old` gelöscht werden. Im Fehlerfall ist ein Einlesen des alten Systems immer noch möglich.

### 3.5. Zusätzliche Schritte

Soll der neu generierte Kern auf einem System eingesetzt werden, dessen bisheriger Kern im anderen Modus lief, sind folgende zusätzliche Schritte vor dem erneuten Systemaufruf erforderlich:

- Mittels `ibblock(8)` sind die Boot-Blöcke umzustellen:  
 Umstellung auf DCP-Modus: `/etc/ibblock -H /dev/hd`  
 Umstellung auf Grundmodus: `/etc/ibblock -W /dev/hd`
- Für den Bildschirmmodus des Editors `em(1)` ist der dem Betriebssystemmodus entsprechende von der ROOT1-Diskette zu nutzen (z.B. durch Kopieren nach `/bin/em`).

Diese Moduswechsel sollten eine Ausnahme bleiben. Sind beide Varianten erforderlich, ist das Einrichten einer zweiten MUTOS-Partition zu empfehlen. Diese Partition kann auf ein Root-File-System beschränkt bleiben und ansonsten die File-Systeme der anderen Partition nutzen. Derartige Arbeitsweisen werden jedoch von den Installationsprozeduren nicht unterstützt und bleiben Spezialisten vorbehalten.

Werden Veränderungen an den Headern in `/usr/sys/h` vorgenommen, sind diese Header nach `/usr/include/sys` zu kopieren.

Die Directory `/usr/src/cmd` ist für die Quellen von Programmen vorgesehen, die vom Anwender unter Umständen modifiziert oder ergänzt werden können. Der folgende Abschnitt beinhaltet ein Beispiel dafür.

### 3.6. Anschluß von Terminals, die nicht zum Systembestand gehören

~~~~~

Sollen Terminals anderen Typs als K 8911 angeschlossen und im Mehrnutzerbetrieb bedient werden, ist das durch Ergänzung der Tabelle im Kommando `getty(8)` möglich. Die Quelle zu diesem Programm befindet sich in `/usr/src/cmd`. Für den neuen Terminaltyp ist ein weiteres Kennzeichen einzuführen, und die Parameter in der zugehörigen Tabelle sind entsprechend einzustellen (siehe `tty(4)`). Das eingeführte Kennzeichen ist im File `/etc/ttys` zu benutzen. Nach Übersetzung von `getty.c` und Kopieren von `getty` nach `/etc` wird der neue Terminaltyp im Mehrnutzeranlauf berücksichtigt.

Voraussetzung für einen derartigen Schritt ist natürlich die prinzipiell Ähnliche Arbeitsweise des neuen Gerätes wie K 8911.

Dazu gehören:

- Asynchronbetrieb,
- Datenbreite 7 oder 8 Bit,
- Paritätsbehandlung beliebig,
- Übertragungsgeschwindigkeiten (Baud) 9600, 4800, 2400, 1200  
600, 300, 150, 75, 50,
- keine Übertragungsprozedur.