

# Aufbau von Betriebssystemen

von

Paul G. Caspers

Mit 54 Abbildungen



Sammlung Götschen Band 7013

Walter de Gruyter

Berlin · New York · 1974

Dipl.-Ing. *Paul G. Caspers*  
ist Mitarbeiter der IBM Deutschland,  
Entwicklung und Forschung,  
und Lehrbeauftragter  
an der Technischen Hochschule Darmstadt

ISBN 3 11 004321 1

©

Copyright 1974 by Walter de Gruyter & Co., vormals G. J. Göschen'sche Verlagshandlung, J. Guttentag, Verlagsbuchhandlung, Georg Reimer, Karl J. Trübner, Veit & Comp., 1 Berlin 30.

Alle Rechte, insbesondere das Recht der Vervielfältigung und Verbreitung sowie der Übersetzung, vorbehalten. Kein Teil des Werkes darf in irgendeiner Form (durch Fotokopie, Mikrofilm oder ein anderes Verfahren) ohne schriftliche Genehmigung des Verlages reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

Printed in Germany.

Satz und Druck: Mercedes-Druck, 1 Berlin 61  
Bindarbeiten: Berliner Buchbinderei Wübben & Co., 1 Berlin 42

## Vorwort

Betriebssysteme sind heute fester Bestandteil jeder Datenverarbeitungsanlage. Ihre Größe schwankt von ein paar tausend bis zu mehreren Millionen Instruktionen. Entsprechend unterschiedlich ist der Aufbau der einzelnen Systeme. In diesem Band soll nun ihre Struktur unter dem Gesichtspunkt der Implementierung untersucht werden. Dabei werden die theoretischen Grundlagen wie Systemaufbau, Verarbeitungsarten, Datei- und Speicherorganisation, Verteilungsprinzipien für Betriebsmittel, Aufbau von Warteschlangen und dgl. als bekannt vorausgesetzt (siehe z. B. Sammlung Göschen „Neuhold, Grundlagen der Betriebssysteme“). Das Schwergewicht liegt hier darauf, wie sich diese Grundprinzipien mittels bekannter Programmier-techniken wie Listen und Tabellen verwirklichen lassen.

Die Beispiele wurden meistens dem Betriebssystem IBM OS/360 nachgebildet. Ihre Aussagen gelten aber nicht nur für dieses System. Dadurch wurde eine geschlossene Darstellung möglich, ohne den Umfang eines Taschenbuches zu sprengen.

Dieser Band enthält die wesentlichsten Teile einer Vorlesung, die ich mehrere Jahre lang an der Universität Karlsruhe bzw. der Technischen Hochschule Darmstadt vor Studenten der Informatik gehalten habe. Ausgehend von der Beschreibung einiger spezieller Hardware-Einrichtungen wird die Verarbeitung von Unterbrechungen als Kern der heutigen Betriebssysteme besprochen. Danach wird in erster Linie nach funktionellen Gesichtspunkten der weitere Aufbau untersucht. Dabei beginnen wir mit dem Datenmanagement und der Ein- und Ausgabesteuerung als weiterem zentralen Teil, der selbst in kleinen Betriebssystemen vorhanden ist. Dies gilt auch für die Ladeprogramme. Den Schluß des Buches bilden Prozeß-, Job- und Systemsteuerung. Das Literaturverzeichnis enthält

im allgemeinen nur generelle Hinweise. Weitergehende Literaturstellen sind den Literaturangaben dieser Werke zu entnehmen.

Böblingen, im Juni 1973

*Paul G. Caspers*

## **Inhalt**

Vorwort . . . . .	3
Inhalt . . . . .	5
Verzeichnis der verwendeten Abkürzungen . . . . .	7
<i>1. Hardware-Einrichtungen zur Unterstützung der Betriebssysteme . . . . .</i>	<i>9</i>
1.1 Die Unterbrechungen . . . . .	9
1.2 Das Programmstatuswort . . . . .	11
1.3 Generelle Verarbeitungsregeln für Unterbrechungen . . . . .	12
1.4 Der Schutz des Überwachers . . . . .	14
<i>2. Allgemeiner Aufbau der Betriebssysteme . . . . .</i>	<i>15</i>
2.1 Die Komponenten eines Betriebssystems . . . . .	15
2.2 Die Residenz eines Betriebssystems . . . . .	18
2.3 Die Systemgenerierung . . . . .	21
2.4 Konventionen bei der Implementierung . . . . .	22
2.5 Die Kommunikation zwischen einzelnen Programmoduln . . . . .	24
<i>3. Die Verarbeitung von Unterbrechungen . . . . .</i>	<i>25</i>
3.1 Der Unterbrechungsverteiler . . . . .	25
3.2 Die Aufgaben der Bedienungsroutinen . . . . .	29
3.3 Implementierungshinweise für die Bedienungsroutinen . . . . .	33
<i>4. Das Datenmanagement . . . . .</i>	<i>34</i>
4.1 Der Aufbau des Datenmanagements . . . . .	34
4.2 Die Organisation des Trägermediums . . . . .	36
4.3 Der Dateisteuerblock . . . . .	39
4.4 Routinen zur Datenübertragung . . . . .	43
4.4.1 Direkte Benutzung eines Kanalprogramms . . . . .	43
4.4.2 Übertragen eines Blockes . . . . .	43
4.4.3 Bereitstellen eines Satzes . . . . .	45
4.5 Routinen für Hilfsfunktionen . . . . .	48
4.5.1 Die OPEN-Routine . . . . .	49
4.5.2 Die CLOSE-Routine . . . . .	49
4.5.3 Die EOv-Routine . . . . .	52
4.6 Weitere E/A-Befehle . . . . .	54
4.6.1 Deklarative Anweisungen . . . . .	54
4.6.2 Imperative Befehle . . . . .	54

5. <i>Die Ein-/Ausgabesteuerung</i> . . . . .	55
5.1 Die Start-E/A-Routine . . . . .	55
5.2 Die Prüfroutinen . . . . .	58
5.3 Die E/A-Unterbrechungsroutinen . . . . .	59
5.4 Die Fehlerkorrekturroutinen . . . . .	59
5.5 Die Verwaltung der E/A-Geräte . . . . .	62
5.6 Der Kommunikationsbereich . . . . .	64
6. <i>Die Prozeßsteuerung</i> . . . . .	65
6.1 Der Prozeßsteuerblock . . . . .	65
6.2 Der Prozeßumschalter . . . . .	66
6.3 Der Aufruf von Programmen . . . . .	70
6.4 Die Speicherplatzverwaltung . . . . .	74
6.5 Die Kommunikation zwischen Prozessen . . . . .	78
7. <i>Das Laden von Programmen</i> . . . . .	80
7.1 Die Aufgabenstellung . . . . .	80
7.2 Die Ausgabe der Sprachübersetzer . . . . .	81
7.3 Einfache Lader . . . . .	83
7.4 Der Binder . . . . .	83
8. <i>Die Jobsteuerung</i> . . . . .	91
8.1 Die Ausbaustufen der Jobsteuerung . . . . .	91
8.2 Die Jobsteueranweisungen . . . . .	95
8.3 Der interne Aufbau der Jobsteuerung . . . . .	97
8.3.1 Der Eingabe-Leser . . . . .	97
8.3.2 Der Prozeßstarter . . . . .	99
8.3.3 Der Prozeßbeender . . . . .	102
8.3.4 Der Ausgabe-Schreiber . . . . .	102
9. <i>Die Systemsteuerung</i> . . . . .	103
Literaturverzeichnis . . . . .	106
Sachregister . . . . .	108

## **Verzeichnis der verwendeten Abkürzungen**

(Die Zahlen in Klammern verweisen auf die Abschnitte, in denen die Abkürzungen zum ersten Mal benutzt werden.)

BM	Bandmarke (4.2)
DAK	Dateianfangskennsatz (4.2)
DAKF	Dateianfangskennsatz, Fortsetzung der Datei (4.2)
DCB	Teil des DSB: data control block (4.3)
DEB	Teil des DSB: data extent block (4.3)
DEK	Dateiendekennsatz (4.2)
DEKF	Dateiendekennsatz, Fortsetzung der Datei (4.2)
DSB	Dateisteuerblock (4.3)
E/A	Ein-/Ausgabe (4.1)
EAWS	Warteschlange für E/A-Anforderungen (5.1)
ECB	Teil des DSB: event control block (4.3)
EOB	Ende des Blockes (end of block) (4.4.3)
EOF	Ende der Datei (end of file) (4.5.3)
EOFSW	Schalter im DSB zur EOVB/EOF-Anzeige (4.5)
EOV	Ende des Stapels (end of volume) (4.5.3)
ESB	Ereignissteuerblock (6.5)
ESID	Externe Symbolidentifizierung (7.2)
EXCP	Ausführung des Kanalprogrammes (execute channel program) (4.4.1)
FQE	Element der Freispeicherliste (free queue element) (6.4)
IOB	Teil des DSB: input/output block (4.3)
IPL	Ladevorgang bei Inbetriebnahme des Systems (initial program loading) (2.2)
NIP	Programm zur Initialisierung des Kerns des Betriebssystems (nucleus initialization program) (2.2)
PIOT	Prozeß-E/A-Tabelle (process input/output table) (8.3.2)
PQE	Element der Speicheraufteilungsliste (partition queue element) (6.4)

PRB	Steuerblock zur Beschreibung eines Programmoduls (program request block) (6.3)
PSB	Prozeßsteuerblock (6.1)
PSW	Programmstatuswort (1.2)
SVC	Supervisoraufruf (supervisor call) (1.1)
TAD	Tabelle der Abschnittsdefinitionen (7.2)
TEP	Tabelle der Eintrittspunkte (7.2)
TES	Tabelle der externen Symbole (7.2)
TVA	Tabelle der zu verschiebenden Adressen (7.2)
TXT	Text, Programmstück (7.2)
UCB	Teil des DSB: unit control block (4.3)
VF	Verschiebungsfaktor (7.4)

# 1. Hardware-Einrichtungen zur Unterstützung der Betriebssysteme

## 1.1 Die Unterbrechungen

Damit die Betriebssysteme die ihnen zugedachten Aufgaben der Betriebsmittelverwaltung und der Arbeitsablaufsteuerung der Datenverarbeitungsanlage durchführen können, benötigen sie spezielle Funktionen in der Hardware. Die wichtigste ist die „Unterbrechung“ (interrupt). Dies ist eine Einrichtung, die es gestattet, den normalen Ablauf eines Prozesses beim Eintreten bestimmter Ereignisse zu unterbrechen und stattdessen einen anderen Prozeß zu starten (Abb. 1.1). Dieser beginnt auf einer fest vorgegebenen Speicherstelle und kann die Anforderungen bedienen, die durch die Unterbrechung gestellt werden. Danach kann der erste Prozeß wieder an der Stelle fortgesetzt werden, an der er unterbrochen wurde.

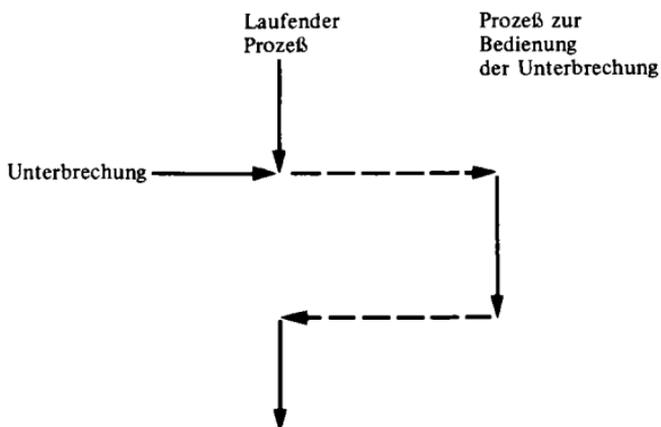


Abb. 1.1: Verarbeiten einer Unterbrechung

Man hat mehrere Klassen von Unterbrechungen entwickelt, die bei verschiedenartigen Ereignissen eintreten. Solche sind z. B. beim IBM-System/360:

1. Maschinenfehler
2. Programmfehler
3. Supervisoraufruf-Instruktionen
4. Externe Signale
5. Ein-/Ausgabeoperationen

Unterbrechungen der ersten Art werden durchgeführt, wenn ein Fehler in der Hardware der Maschine auftritt (z. B. defekter Schaltkreis, defekte Speicherstelle). Programmfehler zeigen Fehler im Programm an, so daß eine Maschineninstruktion nicht ausgeführt werden kann (z. B. Division durch null, ungültiges Vorzeichen). Die Supervisoraufruf-Instruktion (SVC, supervisor call instruction) ist eine spezielle Maschineninstruktion, die es dem laufenden Prozeß gestattet, selbst eine Unterbrechung zu erzeugen, um so Dienstleistungen des Überwachers anzufordern. Externe Signale können durch Betätigen des Unterbrecherknopfes an einer Konsole, durch Ablauf eines Zeitgebers oder auch über Datenfernleitungen in das System kommen. Ein-/Ausgabe-Unterbrechungen treten jedesmal auf, wenn eine Ein- oder Ausgabeoperation angestoßen oder beendet wird, um den Prozessor über das Arbeiten des Kanales und des E/A-Gerätes zu benachrichtigen. Dies sind die häufigsten Unterbrechungen.

Die Zahl der Klassen hängt von der Hardware ab. Sind es beim IBM-System/360 fünf Klassen, so hat das IBM-System/1800 27 verschiedene. Bei unseren weiteren Betrachtungen wollen wir im wesentlichen das IBM-System/360 als Beispiel wählen. Andere Systeme arbeiten ähnlich.

Während der SVC-Befehl vom laufenden Prozeß gegeben wird, geschehen alle anderen Unterbrechungen asynchron zu ihm. Unterbrechungen durch Programmfehler und SVC-Befehl schließen sich gegenseitig aus und können daher nicht gleichzeitig auftreten, wohl aber alle anderen. Deshalb müssen Priori-

täten vorgegeben werden, in welcher Reihenfolge die Unterbrechungen angezeigt werden sollen, wenn mehrere zur gleichen Zeit auftreten. Eine solche Reihenfolge kann z. B. obige Rangordnung sein.

## 1.2 Das Programmstatuswort

Bei einer Unterbrechung wird die Kontrolle über die Zentraleinheit einem anderen Prozeß übergeben. Bei einem solchen Wechsel zwischen verschiedenen Prozessen wird jedesmal der Zustand der Zentraleinheit verändert. Soll ein durch eine Unterbrechung unterbrochener Prozeß später wieder fortgesetzt werden, so muß erst der Zustand in der Zentraleinheit wiederhergestellt werden, wie er vor der Unterbrechung bestanden hat. Man faßt daher gerne alle Größen, die den Zustand der Zentraleinheit beschreiben, in einem *Programmstatuswort* (PSW) zusammen. Zu diesen Größen gehören z. B. der Befehlszähler, der Speicherschutzschlüssel (siehe Abschnitt 1.4), Maskierungen für Unterbrechungen (siehe Abschnitt 1.3) und Statusanzeiger wie das Ergebnis einer Vergleichsoperation u. ä. (Abb. 1.2).

Für einen laufenden Prozeß steht das PSW in einem internen Register des Prozessors. Tritt eine Unterbrechung ein, so wird es automatisch aus diesem internen Register auf einen definierten Platz des Hauptspeichers abgesetzt und stattdessen ein

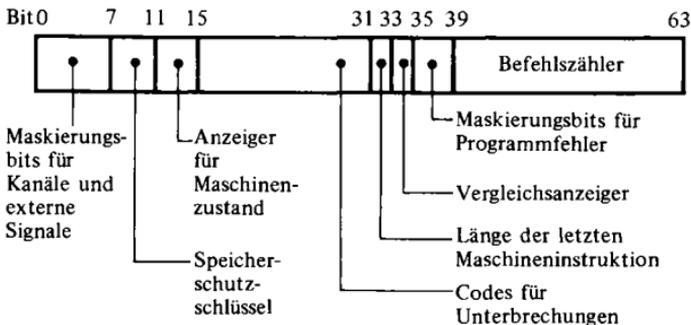


Abb. 1.2: Programmstatuswort des IBM-Systems/360

## 12 1. Hardware-Einrichtungen zur Unterstützung der Betriebssysteme

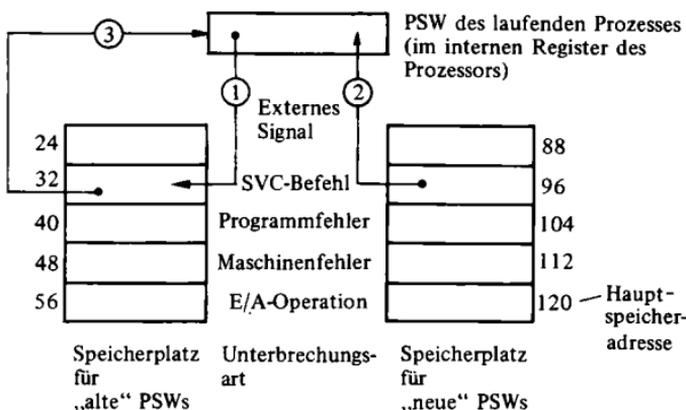


Abb. 1.3: Austausch der PSWs im IBM-System/360

Dargestellt ist das Eintreten einer SVC-Unterbrechung:

- 1 Das PSW des lfd. Prozesses wird auf den Platz für alte SVC-PSWs abgespeichert (automatisch)
- 2 Das PSW für den die Unterbrechung verarbeitenden Prozeß wird von dem Speicherplatz für neue SVC-PSWs in interne Register geladen (automatisch)
- 3 Nachdem der neue Prozeß die Unterbrechung bedient hat, ladet er mit einer speziellen Instruktion (Load PSW) das PSW des ersten Prozesses vom Speicherplatz für alte SVC-PSWs in das interne Prozessorregister zurück, um den ersten Prozeß wieder fortzusetzen

anderes PSW, das zu dem die Unterbrechung bedienenden Prozeß gehört, in das interne Register geladen. Will man später wieder zu dem alten Prozeß zurückkehren, muß das zu diesem Prozeß gehörige PSW mittels spezieller Instruktionen (z. B. „Load PSW“) vom laufenden Prozeß wieder geladen werden (Abb. 1.3).

### 1.3 Generelle Verarbeitungsregeln für Unterbrechungen

Um eine Unterbrechung verarbeiten zu können, muß zunächst verhindert werden, daß eine zweite Unterbrechung zum Minde-