

<b>Zeitschrift:</b>	Bulletin des Schweizerischen Elektrotechnischen Vereins, des Verbandes Schweizerischer Elektrizitätsunternehmen = Bulletin de l'Association suisse des électriciens, de l'Association des entreprises électriques suisses
<b>Herausgeber:</b>	Schweizerischer Elektrotechnischer Verein ; Verband Schweizerischer Elektrizitätsunternehmen
<b>Band:</b>	65 (1974)
<b>Heft:</b>	10
<b>Artikel:</b>	Betriebssysteme für Prozessautomatisierung
<b>Autor:</b>	Nemetz, P.
<b>DOI:</b>	<a href="https://doi.org/10.5169/seals-915405">https://doi.org/10.5169/seals-915405</a>

### **Nutzungsbedingungen**

Die ETH-Bibliothek ist die Anbieterin der digitalisierten Zeitschriften auf E-Periodica. Sie besitzt keine Urheberrechte an den Zeitschriften und ist nicht verantwortlich für deren Inhalte. Die Rechte liegen in der Regel bei den Herausgebern beziehungsweise den externen Rechteinhabern. Das Veröffentlichen von Bildern in Print- und Online-Publikationen sowie auf Social Media-Kanälen oder Webseiten ist nur mit vorheriger Genehmigung der Rechteinhaber erlaubt. [Mehr erfahren](#)

### **Conditions d'utilisation**

L'ETH Library est le fournisseur des revues numérisées. Elle ne détient aucun droit d'auteur sur les revues et n'est pas responsable de leur contenu. En règle générale, les droits sont détenus par les éditeurs ou les détenteurs de droits externes. La reproduction d'images dans des publications imprimées ou en ligne ainsi que sur des canaux de médias sociaux ou des sites web n'est autorisée qu'avec l'accord préalable des détenteurs des droits. [En savoir plus](#)

### **Terms of use**

The ETH Library is the provider of the digitised journals. It does not own any copyrights to the journals and is not responsible for their content. The rights usually lie with the publishers or the external rights holders. Publishing images in print and online publications, as well as on social media channels or websites, is only permitted with the prior consent of the rights holders. [Find out more](#)

**Download PDF:** 13.01.2026

**ETH-Bibliothek Zürich, E-Periodica, <https://www.e-periodica.ch>**

# Betriebssysteme für Prozessautomatisierung

Von P. Nemetz

Neben der Grund-Software (Assembler, Compiler, usw.) werden heute mit den Prozessrechnern auch Betriebssysteme für die Behandlung von Echtzeitaufgaben (Real Time Applications) abgegeben. Sie unterstützen die Entwicklung der Anwendungsprogramme, da sie die prozessunabhängigen Aufgaben behandeln. Durch entsprechende Erweiterungen erreicht man, dass ihre Anwendung in der Netzautomatisierung die Programmerstellung vereinfacht. Die Grösse des Rechnersystems legt dabei fest, wie weitgehend solche Ausbauten eines Betriebssystems tatsächlich durchgeführt werden können.

62-503.55 : 681.3-9

Maintenant, les calculateurs de processus sont livrés non seulement avec toutes les instructions nécessaires (assemblage, compilation, etc.), mais également avec des systèmes de traitement des applications en temps réel, qui permettent le développement de programmes, car ils se rapportent à des tâches indépendantes des processus. Par des extensions appropriées, l'établissement des programmes d'automatisation des réseaux est nettement simplifié. La grandeur du système de calculateur détermine dans quelle mesure ces extensions d'un système d'exploitation sont effectivement possibles.

## 1. Einleitung

Unter dem Begriff «Betriebssystem» (Real Time Executive) versteht man ein Programmssystem, das die Entwicklung der anlagen spezifischen Software unterstützt. Es wird, da es unabhängig von den jeweiligen Anwendungen bei der Prozessautomatisierung ist, vielfach von den Rechnerfirmen entwickelt und entweder als Softwaresupport mit der Hardware abgegeben oder – als Folge der immer stärker ins Gewicht fallenden Entwicklungskosten – als eigenes Produkt verkauft. Bei der Wahl eines Rechners spielt neben vielen anderen Faktoren auch das zur Verfügung stehende Betriebssystem eine Rolle. Ein Optimum kann nur gefunden werden, wenn man trachtet, möglichst viele der zum Teil widersprüchlichen Gesichtspunkte zu berücksichtigen:

– **Aufgabenstellung:** Die Festlegung der Aufgabenstellung ist vielfach schwierig, weil zum Teil die Unterlagen dazu benötigt werden, die man mittels des geplanten Systems erfassen will. Es besteht der Wunsch, Betriebserfahrungen direkt zur Verbesserung der Systemführung anwenden zu können.

– **Einsatzmöglichkeiten:** Für die Prozessführung ist im Durchschnitt wenig Rechenzeit erforderlich. Die Ausnutzung der restlichen Zeit wird mit steigenden Anlagekosten vermehrt eine Frage der Wirtschaftlichkeit, erfordert aber andererseits weitere Investitionen.

– **Systemumfang:** Bei grossen Netzen werden zum Teil schon in den Unterwerken Prozessrechner eingesetzt. In den übergeordneten Zentren werden vielfach andere Rechner verwendet, was zu Problemen beim Unterhalt, der Ersatzteilbeschaffung und der Programmierung führt.

– **Integration:** Der Prozessrechner ist immer nur ein Teil eines Gesamtsystems. Erst das reibungslose Zusammenspiel aller Komponenten (Fernwirkanlagen, örtlichen Steuerungen, Kommandostationseinrichtungen, Software, usw.) kennzeichnet ein gut aufgebautes System.

– **Personal:** Die normale Betriebsführung erfolgt in den meisten Fällen durch Personal, das dem Rechner fremd gegenübersteht. Daher muss der Bedienungskomfort hoch sein, um einen sicheren Betrieb zu garantieren [1]). Diese Tatsache, ebenso wie gute Schulungs- und Trainingsmöglichkeiten verlangen umfangreichere Systeme.

– **Sicherheit, Verfügbarkeit:** Hohe Anforderungen dieser Art sind einer der wesentlichsten Gründe für hohe Systemkosten. Der Ausfall einzelner Komponenten darf den Betrieb des Gesamtsystems nicht oder nur unwesentlich beeinträchtigen (Verdopplung wichtiger Einrichtungen, Prüfeinrichtungen, Möglichkeit der Umschaltung von Übertragungswegen, usw.) [2].

– **Wirtschaftlichkeit:** Diese Frage ist ausserordentlich schwierig zu beantworten, da billige Systeme nicht unbedingt wirtschaftlich sind. Erhöhte Sicherheit durch Komponentenvervielfachung oder durch technisch hochwertigere Systeme, höherer Bedienungskomfort, einfachere Entwicklung von Software, usw. sind Punkte, die sich im allgemeinen nicht in Geld ausdrücken

lassen, im Extremfall aber über den Wert oder Sinn einer Anlage entscheiden.

Bei diesem kurz umrissenen Problemkreis nimmt das Betriebssystem als Grundlage der Software-Entwicklung insoweit eine wichtige Position ein, als es bei der Rechnerwahl mitberücksichtigt werden sollte, auch wenn es mit seinen Möglichkeiten ziemlich stark von der zur Verfügung stehenden Grösse des Prozessrechners abhängig ist. Ausgehend von einer kurzen Diskussion der Einsatzmöglichkeiten von Prozessrechnern in elektrischen Netzen und einigen Bemerkungen zur Entwicklung von Software werden im folgenden die wichtigsten Funktionen der Betriebssysteme erklärt. Daran anknüpfend wird gezeigt, welche Erweiterungsmöglichkeiten für die Fernwirksysteme und im besonderen für die Netzautomatisierung sich ergeben.

## 2. Der Einsatz von Prozessrechnern in elektrischen Netzen

Den folgenden Überlegungen wird als Modell für ein elektrisches Netz ein System mit hierarchischer Struktur [3] zugrunde gelegt (Fig. 1). Diese Annahme ist eine praktisch immer anwendbare Verallgemeinerung. In Tabelle I ist eine Auswahl von Aufgaben für die einzelnen Ebenen zusammengestellt. In den höheren Hierarchie-Stufen ist der

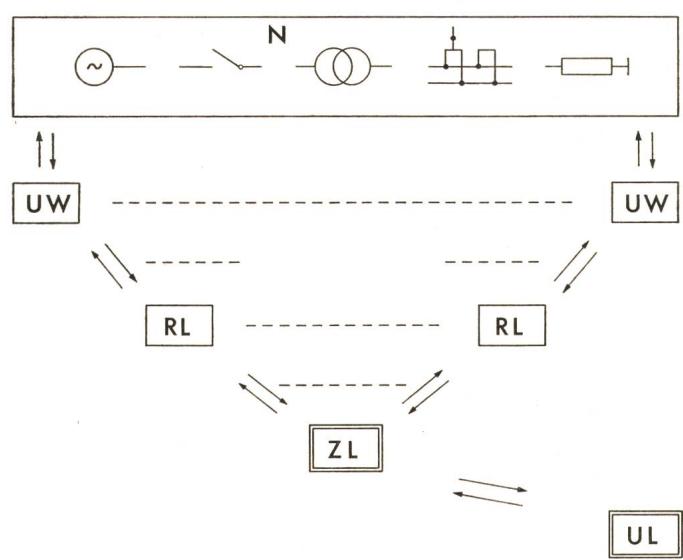


Fig. 1 Struktur einer Netzaufschaltung als hierarchisches System

N zu überwachendes Netz

UW Unterwerke

RL regionale Leitstelle

UL, ZL übergeordnete Leitstelle

1) Siehe Literatur am Schluss des Aufsatzes.

Hierarchie-Ebenen	Unterwerk (UW) (Kraftwerk, Schaltstation usw.)	Leitstelle Regionales Zentrum (RL)	Zentrale Leitstelle Nationales Zentrum (NZ) (ZL)	Übergeordnete Leitstelle Internationales Zentrum (IZ) (UL)	
Typische Aufgaben- stellung	Primär- und Sekundär- regelung Steuerung von Schaltern, Trennern usw. Anlagenschutz Protokollierung Verbindung mit dem (RL)	Protokoll Verbindung mit UW und dem ZL Abrechnung (Zählerstände) eventuell: LFC, EDC LFC, eventuell EDC Statistik	Protokollierung Verbindung mit RL und dem UL Abrechnung LFC, EDC Statistik Übergeordneter Netzschutz Netzsicherheit Einsatzplanung von Kraftwerken Projektierung von Anlagen	Protokollierung Verbindung mit ZL Abrechnung der Austauschleistungen LFC für gleichberechtigte Partner Statistik Übergeordneter Netzschutz für Projektierung von Anlagen Beratung, Koordination	
Kleinste- und Kleinrechnersysteme			Mittelgrosse und grosse Rechnersysteme		
Rechnertyp Betriebssysteme (typischer Speicherbedarf)	(Einzweckgerät)	0,5...1 k	1...3 k	4...6 k	5...20 k

LFC = Leistungsfrequenzregelung; EDC = ökonomische Lastverteilung.

Einsatz eines Rechners immer mehr gerechtfertigt. Dementsprechend ist ein Übergang von kleineren zu grösseren Prozessrechnern festzustellen, ebenso wie eine Verlagerung des Charakters der Aufgaben von den Prozessanwendungen zur Datenverarbeitung.

In den Unterwerken findet man im allgemeinen Kleinrechner, die für einen relativ kleinen und gut definierbaren Aufgabenbereich eingesetzt werden [4; 5; 6]. Die hier verwendeten Betriebssysteme [7] sind einfach und der Rechner hat die Eigenschaften eines Einzweckgerätes. In den regionalen Leitstellen können auch grössere Rechnersysteme entsprechend den erweiterten Aufgabenstellungen zur Anwendung kommen. Je nach der Struktur des Gesamtsystems kann die Programmentwicklung sowohl gemeinsam für mehrere solcher Zentren in einem entsprechend ausgerüsteten Rechenzentrum, als auch individuell mit Hilfe der – im allgemeinen dann grösseren – Prozessrechner erfolgen. In den zentralen und übergeordneten Leitstellen kommen nur mehr grössere Prozessrechnersysteme [8] oder Datenverarbeitungsanlagen in Frage, die für die Verarbeitung der umfangreichen Programme und grossen Datenmengen geeignet sind.

### 3. Betriebssysteme

Der Begriff des Betriebssystems kommt von den Datenverarbeitungsanlagen. Man versteht darunter ein Programm- system zur Steuerung des Ablaufes einer Reihe von Programmen in einem Rechner mit dem Ziel, dessen Preisleistungsverhältnis möglichst günstig zu gestalten [10]. Beim Prozessrechner stehen andere Aufgaben im Vordergrund, doch

kommt aus wirtschaftlichen Gründen bei den grossen Anlagen die bessere Ausnutzung der zur Verfügung stehenden Rechenzeit als Nebenbedingung hinzu.

Bei Programmsystemen für Prozessanwendungen lässt sich fast immer eine Aufteilung gemäss Fig. 2 durchführen. Die Software-Entwicklung beschränkt sich bei den mittleren und grossen Systemen auf die Erstellung der Anwendeprogramme und Teile der Unterprogrammbibliothek. Bei kleinen Systemen ist diese Trennung nur zum Teil möglich, da – um den Speicherbedarf zu verringern – Betriebssystem, Unterprogramme und Anwenderprogramme enger miteinander verknüpft sind.

Im folgenden werden die wichtigsten Funktionen eines Betriebssystems zusammengestellt und kurz erläutert. Je nach Grösse des Prozessrechners kann ein einfaches oder komplexes Betriebssystem verwendet werden, das einen mehr oder weniger grossen Teil der eingangs erwähnten Funktionen beinhaltet. Die Eigenschaften dieser Betriebssysteme werden einander gegenübergestellt und anschliessend daran wird auf die Unterprogrammbibliothek und den Begriff der Systemgenerierung eingegangen.

#### 3.1 Die wichtigsten Funktionen eines Betriebssystems

Im folgenden werden einige Techniken beschrieben, die die Basis der Betriebssysteme bilden:

##### 3.1.1 Die Interrupt-Behandlung (Interrupt Handling):

Interrupts erzwingen die Unterbrechung des gerade aktiven Programms, und es erfolgt ein Sprung auf eine bestimmte Adresse im Kernspeicher, wo die Startadresse des

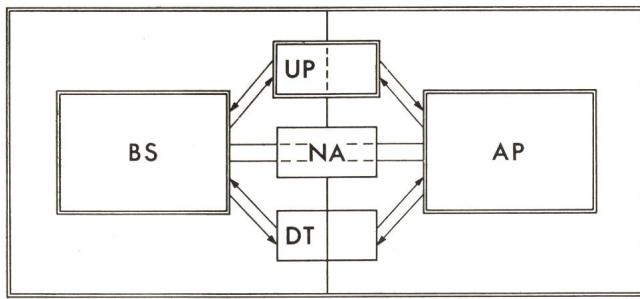


Fig. 2 Prinzipieller Aufbau eines Programmsystems für Prozessrechner

Der linke Teil ist unabhängig von der jeweiligen Problemstellung, der rechte deren Lösung

BS Betriebssystem  
AP Anwendungsprogramme  
UP Unterprogrammbibliothek  
NA normierte Anrufe  
DT Datentabellen

sog. Interrupt-Behandlungsprogrammes (Interrupt Response Routine) eingetragen ist. Vor Inangriffnahme der eigentlichen Interrupt-Behandlung wird die richtige Fortsetzung des unterbrochenen Programms sichergestellt (Housekeeping). Bei Auftreten mehrerer Interrupts gleichzeitig wird stets der mit der höchsten Priorität zuerst bearbeitet, bei der Behandlung eines Interrupts werden alle weiteren niedrigeren Priorität verzögert, während solche mit höherer Priorität bedient werden (Nesting of Interrupts). Wichtige Aufgaben, die möglichst unverzögert behandelt werden müssen, sind daher stets mit einem Interrupt verknüpft. Unter «möglichst unverzögert» versteht man Zeiten von 800 ns bis einige Millisekunden, je nach Rechnerprodukt, Priorität der Aufgabe und Betriebssystem. Interrupt-Behandlungsprogramme werden immer möglichst kurz gehalten, um die niedrigeren Prioritäts-ebenen (Levels) nicht zu blockieren. Sie führen daher nur die Funktionen aus, die strengen Zeitbedingungen unterliegen, während die anderen nicht mehr auf dem Interrupt-Niveau bearbeitet werden.

### 3.1.2 Die Warteschlangentechnik (Queuing Technique, Program Scheduling)

Diese Technik sieht die Ausführung einzelner Aufgaben entsprechend ihrer Priorität und – bei grösseren Systemen vor allem – bei gleicher Priorität in der zeitlichen Reihenfolge ihrer Anforderungen vor. Eine neue Aufgabe wird erst begonnen, wenn die vorhergehende abgeschlossen ist, eine Unterbrechung ist nur durch Interrupts möglich. Aufgaben mit grossen Ausführungszeiten müssen so bearbeitet werden, dass sie den Rechner nicht blockieren können.

### 3.1.3 Periodische Aufgaben

Periodische Aufgaben sind häufig anzutreffen. Die Zeit wird im Rechner mit Hilfe eines Gebers (Timer, Real Time Clock) gebildet, der mit einer konstanten Frequenz Interrupts auslöst, die vom Betriebssystem zur Zeitzählung herangezogen werden. Die Bildung der Uhrzeit ist in jedem Betriebssystem anzutreffen (beim Start wird die aktuelle Zeit eingegeben), seltener hingegen die Bildung des Datums. Die Ausführung von periodischen Aufgaben ist so vorbereitet, dass man den Programmnamen und die gewünschte Periode vorgibt. Dieses Programm wird nach Ablauf der vorgegebenen Zeit entweder in die Warteschlange eingetragen oder als Interrupt-Programm mit bestimmter Priorität ausgeführt.

### 3.1.4 Programmierte Interrupts

Damit gewinnt man die Möglichkeit, von Programmen aus Interrupts auszulösen und die sofortige Ausführung wichtiger Aufgaben zu erzwingen.

### 3.1.5 Der Austausch von Kernspeicherbereichen (Core Exchange)

Dieser kommt bei grösseren Rechnern zur Anwendung, wenn neben dem Kernspeicher zum Beispiel ein Plattspeicher zur Verfügung steht. Ein Bereich des Kernspeichers kann dann von mehreren auf der Platte verspeicherten Programmen benutzt werden, indem man bei Bedarf das jeweilige Programm lädt und ausführt.

Die zuletzt angegebenen Beispiele setzen grössere Betriebssysteme voraus. Man findet hier eine Vielzahl von weiteren Funktionen, die vor allem die Software-Entwicklung erleichtern.

## 4. Betriebssysteme für kleinere Rechnerkonfigurationen [7]

Im wesentlichen bestehen sie aus den ersten der drei obengenannten Komponenten und weisen daher einen geringen Kernspeicherbedarf auf (typisch sind 0,5...2 k), so dass sie auch bei sehr kleinen Konfigurationen eingesetzt werden können. Im allgemeinen ist neben dem Interrupt-Behandlungsprogramm für den Timer auch das für eine Schreibmaschine (Konversationsschreibmaschine) enthalten, über die Anweisungen an das Betriebssystem (Aufrufen von Programmen, Ändern von Werten im Speicher, Drucken bestimmter Speicherbereiche usw.) «on line» eingegeben werden können. Für standardmässige Peripheriegeräte werden die entsprechenden Behandlungsprogramme abgegeben, die an das Betriebssystem angeschlossen werden können.

Im einfachsten Fall erfolgt der Zusammenbau eines Programmsystems folgendermassen: Das Betriebssystem und alle erforderlichen Programme werden, nachdem sie entwickelt, übersetzt und getestet wurden, in den Kernspeicher geladen und ihre Startadressen und Namen in Tabellen des Betriebssystems eingetragen. Im Minimum stehen eine Tabelle für die Interrupt-Behandlungsprogramme und eine für die nach dem Warteschlangenprinzip behandelten Programme zur Verfügung.

Bei grösseren Rechnersystemen ist das Betriebssystem weiter ausgebaut. Wenn beispielsweise ein Plattspeicher vorhanden ist, kann dieser nicht nur als Daten-, sondern auch als Programmspeicher benutzt werden. Der Kernspeicher weist dann einen Bereich auf, der abwechselnd von verschiedenen Programmen benutzt werden kann. Das ergibt eine bessere Ausnützung des teuren Kernspeichers, wobei aber nur Aufgaben dafür in Frage kommen, die keinen strengen zeitlichen Bedingungen unterliegen.

So gesehen zeichnet sich bereits ein Übergang zu den grösseren Systemen ab. Sofern sie Magnetplatten als Programmspeicher benutzen, kann man als ihr Charakteristikum die bessere Ausnützung des Kernspeichers bezeichnen. Gemeinsam für alle Betriebssysteme dieser Art ist die Tatsache, dass der Rechner, sobald er an den Prozess angeschlossen (on line) ist, zu einem Einzweckgerät wird. Sollen Programme geändert oder neu entwickelt werden, so muss dafür der Rechner ausser Betrieb (off line) genommen werden.

## 5. Betriebssysteme für grosse Rechnerkonfigurationen [10]

Im Sinne dieser Klassifikation werden die Betriebssysteme als «gross» angesprochen, die sowohl die Prozessführung (foreground jobs), wie auch prozessunabhängige Aufgaben (background jobs) gestatten. Neben der Mehrfachausnutzung des Kernspeichers, die wegen der umfangreichen Organisations- und Hilfsprogramme zu einer Notwendigkeit wird, ermöglichen diese Systeme auch die Ausnutzung der nicht für die Prozessführung benötigten Zeit. Der grosse Vorteil dieser Betriebssysteme ist durch die Tatsache gegeben, dass das Übersetzen, Laden und Testen von Programmen «on line» möglich ist. Dies setzt aber einen Minimalkauf des Rechners voraus, der bedeutend grösser ist als im vorgehenden Fall.

Diese Betriebssysteme kommen vor allem dann zum Einsatz, wenn umfangreichere Aufgaben vorliegen, die Anwendungsprogramme erst mit den Betriebserfahrungen fertiggestellt werden können, während des Betriebes umfangreiche Änderungen zu erwarten sind oder wenn ein erheblicher Anteil von prozessunabhängiger Arbeit mit dem gleichen Rechner durchzuführen ist. Die Entwicklungszeit für diese Betriebssysteme ist ungleich grösser als für die zuerst beschriebenen. Dies erklärt auch die Tatsache, dass, verglichen mit der Zahl der Rechnerfabrikate, wenig Betriebssysteme dieser Art heute auf dem Markt sind.

## 6. Unterprogrammbibliothek, Systemgenerierung

Die Unterprogrammbibliothek ist ein wichtiger Teil der anwenderunabhängigen Software. Sie besteht im allgemeinen aus den Ein/Ausgabeprogrammen zur Bedienung der Standard-Peripheriegeräte des Rechners (Lochstreifen- und Lochkartengeräte, Schreibmaschinen, Magnetplatten usw.), den mathematischen Routinen (Gleitkommaarithmetik, Win-

kelfunktionen usw.) und den Unterprogrammen für FORTRAN.

Um den Kernspeicher optimal auszunützen zu können, werden von allen zur Verfügung stehenden Unterprogrammen nur jene verspeichert, die vom Betriebssystem und den Anwendungsprogrammen gebraucht werden. Entsprechend seinem allgemeinen Charakter ist das Betriebssystem so konzipiert, dass es alle möglichen Standard-Peripheriegeräte eines Prozessrechnersystems bedienen kann. Bevor es für eine bestimmte Aufgabe eingesetzt werden kann, muss es zunächst auf die entsprechende Hardwarekonfiguration zugeschnitten werden. Das heißt, es werden alle nicht benötigten Unterprogramme entfernt, Tabellen verkürzt usw., um seinen Speicherbedarf zu minimieren. Diese Aufgabe bezeichnet man als Systemgenerierung. Sie muss vor dem Aufbau der Prozess-Software durchgeführt werden und ist um so komplizierter, je komplexer ein Betriebssystem ist.

## 7. Entwicklungstendenzen

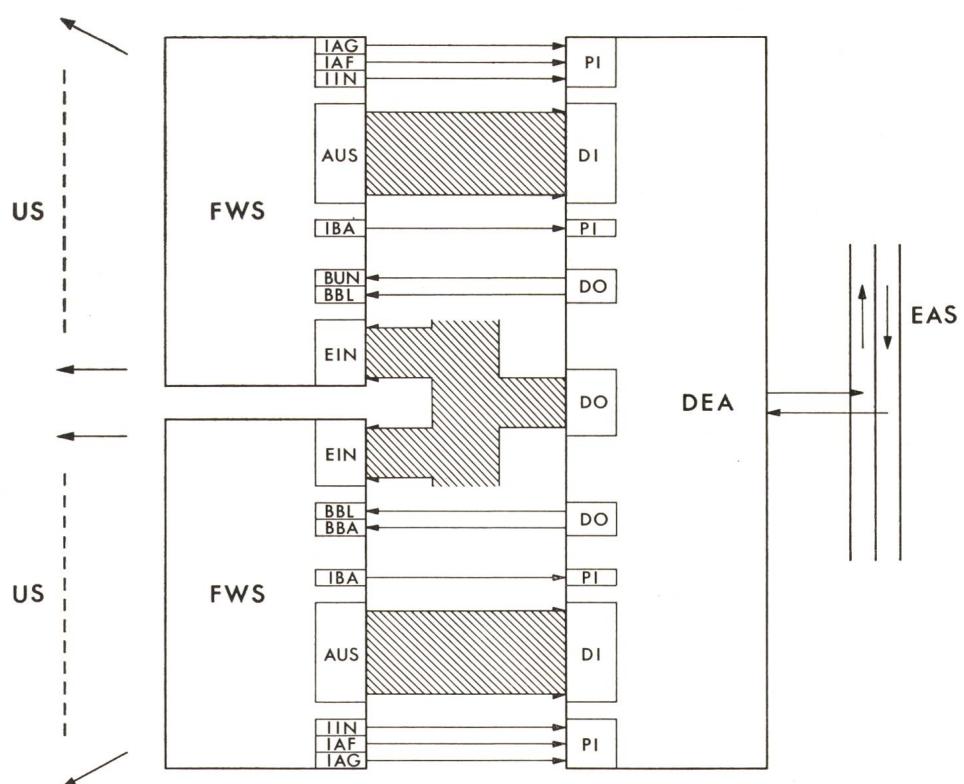
Die fallenden Preise der Hardware-Komponenten ermöglichen die Entwicklung von Zentraleinheiten, die neben einem grösseren Instruktionsvorrat auch bestimmte Aufgaben der Betriebssysteme übernehmen können. Das erleichtert die Entwicklung der Software ganz allgemein und führt zu einer besseren Auslastung der Rechner.

Neben der grösseren Zahl von Instruktionen wird auch die Zahl der Rechenregister erhöht. Wegen der vielen Möglichkeiten wird die Entwicklung von Software in mancher Beziehung komplizierter, andererseits werden die einzelnen Programme aber kürzer, übersichtlicher und daher auch einfacher zu testen.

Für die Interrupt-Behandlung werden im folgenden zwei heute anzutreffende Lösungen angedeutet: In einem Fall werden die Rechenregister so oft vorgesehen, als Prioritäts-

Fig. 3  
Beispiel einer Nahtstelle zwischen einem Prozessrechner und einem Fernwirksystem

US	Unterstationen
FWS	Fernwirksystem
IAG, IAF	Alarne
AUS	codierte Information (Telegramm)
IIN	AUS darf gelesen werden
IBA	EIN ist vom FWS übernommen worden
BUN	Befehlsübernahme
BBL	Befehlsblockierung
EIN	Befehlscode
EAS	Ein/Ausgabeschiene des Prozessrechners
DEA	Digitales Ein/Ausgabesystem
PI	Interrupts
DI	digitale Eingänge
DO	digitale Ausgänge



ebenen vorhanden sind. Bei einem Interrupt höherer Priorität halten die gerade benutzten Register ihre Inhalte und die Interrupt-Behandlung erfolgt mit den der höheren Ebene zugeordneten Registern. Die Fortsetzung des unterbrochenen Programms kann unmittelbar nach Beendigung der Aufgaben höherer Priorität erfolgen. – Bei der anderen Lösung werden die Adressen der nächsten auszuführenden Instruktion und der Status der Zentraleinheit in eine Puffertabelle (Stack) gerettet. Nach Abschluss des Interrupt-Behandlungsprogramms wird, ebenfalls per Hardware, die letzte Eintragung im Puffer zur automatischen Fortsetzung des unterbrochenen Programms benutzt. Diese Puffertabelle wird daher oft «Last In First Out-Table» genannt und ermöglicht auch das Schachteln (Nesting) von Interrupts.

## 8. Betriebssysteme für Netzautomatisierung

Sieht man von den Anwendungen in den Unterwerken ab, wo man in das Gebiet der Lokalautomatisierung [11; 12] kommt, so arbeiten die Prozessrechner praktisch immer mit Fernwirkgeräten zusammen. Letztere übernehmen die Aufgaben des Informationsaustausches mit den einzelnen Unterstationen und zumindest einen Teil der Informationsdarstellung in der Zentrale. Die Verbindung Fernwirkgerät-Prozessrechner erfolgt über eine eigene Nahtstelle, an welcher die im Fernwirksystem zu übertragende, redundanzfreie Information in codierter Form übergeben wird. Ein Telegramm wird jeweils solange an der Nahtstelle gehalten, bis das nächste zur Verfügung steht. Das bedeutet, dass das Fernwirkgerät den Rhythmus der Informationsübergabe bestimmt.

Abgesehen von sehr einfachen Aufgabenstellungen (z. B. nur Protokollierung) wird das Eintreffen jeder neuen Information durch ein Interruptsignal dem Rechner signalisiert. Fig. 3 zeigt das Blockschaltbild einer solchen Nahtstelle, Fig. 4 ein Beispiel eines zeitlichen Verlaufs der Signale. Die Frequenz der Informationsübergabe hängt in erster Linie von der Übertragungsgeschwindigkeit im Fernwirksystem ab. Bei höheren Geschwindigkeiten (z. B. 600, 1200 Baud) und mehreren angeschlossenen Fernwirkgeräten steigt die Belastung des Rechners so, dass Datenkanäle vorgesehen werden müssen, die das Einlesen der Information in den Kernspeicher asynchron zum Programmablauf gestatten (Direct Memory Access, Cycle Stealing). Vom Rechner aus

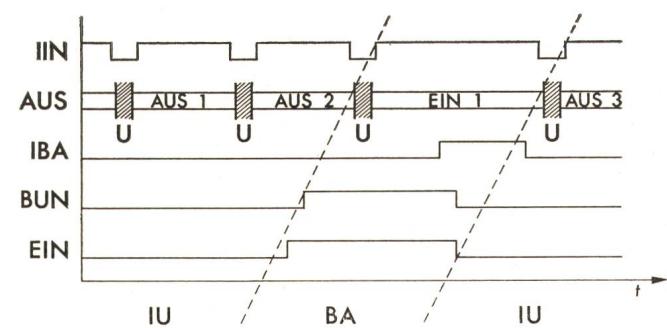


Fig. 4 Beispiel für einen Signalverlauf an der Nahtstelle Fernwirksystem-Prozessrechner

IU Informationsübergabe an den Rechner  
BA Befehlsabsteuerung über das Fernwirksystem  
U Unterbrechung

Weitere Bezeichnungen siehe Fig. 3

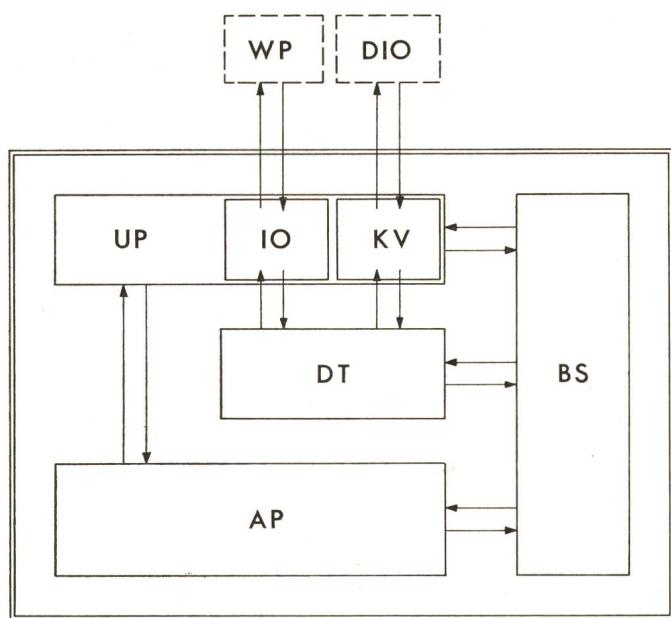


Fig. 5 Aufbau eines Software-Systems für Netzautomatisierung

DIO, KV Digitaler Ein/Ausgang zum Fernwirksystem, zugehörige Programme

WP, IO Weitere Peripheriegeräte, zugehörige Programme

BS Betriebssystem

AP Anwendungsprogramme

UP Unterprogrammbibliothek

DT Datentabellen

gesehen ist das Fernwirkgerät nichts anderes wie ein sehr grosses und komplexes Peripheriegerät. Seine programmtechnische Behandlung kann daher verallgemeinert und der Grund-Software (Betriebssystem und Unterprogrammbibliothek) angegliedert werden. Dabei muss aber berücksichtigt werden, dass die Art der Datenübernahme (per Interrupt oder per Datenkanal) und die Zahl der Fernwirkgeräte in irgendeiner Form berücksichtigt werden muss. Je universeller man diese Lösung aufbaut, um so grösser ist der zu investierende Aufwand an Software-Entwicklung.

Der Einsatz von Programmen, die den Verkehr mit den Fernwirkgeräten übernehmen, erleichtert die Entwicklung der Anwenderprogramme. Die über das Fernwirkgerät eingetroffenen Informationen werden analysiert und in verschiedene Tabellen abgespeichert, auf die alle anderen Programme zugreifen (Fig. 5). Dabei sind einige Punkte zu berücksichtigen:

- Die Anzahl der verschiedenen Informationsarten, die in einem Fernwirksystem vorkommen kann, ist relativ gross (Meldungen, Alarne, Messwerte, Zählerstände, usw.). Für nicht vorkommende Informationsarten darf auch kein Speicherplatz reserviert werden.

- Diese Programme werden so häufig durchlaufen, dass sie stets im Kernspeicher untergebracht und in Assembler geschrieben sind.

- Fernwirksysteme arbeiten selbst mit Adressen (Unterstationenadresse, Messwertadresse, usw.), deren Verwendung zu einfachen Programmen und leicht erweiterbaren Tabellen, aber auch zu einer schlechten Speicherausnutzung, die durch eine flexible Tabellenorganisation verbessert werden muss, führt.

- Die Abspeicherung in diese Tabellen allein ist nur für Messwerte, Zählerstände u. ä., ausreichend, da sie von periodisch aufgerufenen Programmen weiterverarbeitet werden (Grenzwertkontrolle, periodische Protokollierung, Abrechnungen, usw.). Bei

spontan anfallenden Informationen (Meldungen, Alarme, usw.) müssen außerdem Anwenderprogramme aufgerufen werden können (Protokollierung, Schalthandlungen, usw.).

Neben diesen Programmen, die spezifische Aufgaben der Fernwirktechnik behandeln, sind auch solche einzubauen, die verschiedene Anwendung in der Netzautomatisierung betreffen. Hier ist vor allem an übergeordnete Aufgaben Leistung-Frequenz-Regelung, ökonomische Lastverteilung, Sicherheit, State Estimation, usw. gedacht.

Nach dem Obengesagten ist leicht einzusehen, dass die Verallgemeinerung dieser Aufgaben erhebliche Schwierigkeiten mit sich bringt, da die Zahl der zu beherrschenden Möglichkeiten gross ist. Abschliessend werden drei Lösungsmöglichkeiten einander gegenübergestellt.

## 9. Spezialisierungen für Sonderfälle

Sie kommen vor allem in Betracht, wenn kleine Rechnerkonfigurationen zur Verfügung stehen. Man verzichtet auf schon vorhandene verallgemeinerte Lösungen und realisiert ein auf den Anwendungsfall zugeschnittenes optimiertes Programmssystem. Dem Vorteil geringerer Hardwarekosten stehen der höhere Aufwand bei der Programmierung und Schwierigkeiten bei späteren Systemausbauten gegenüber.

## 10. Erweiterung der Unterprogrammbibliothek

Eine relativ einfache Lösung ist die Erweiterung der Unterprogrammbibliothek ohne Änderung des Betriebssystems (entsprechend Fig. 2). Die Aufgaben bis zum Einspeichern der übertragenen Informationen in die verschiedenen Tabellen werden in Form von Unterprogrammen gelöst (Einlesen über die digitalen Eingänge, Parity-Prüfung, Auswertung von Inkrementen bei Zählerstandsübertragung, Ausgabe von Befehlen über das Fernwirksystem, usw.). Für den jeweiligen Anwendungsfall besteht dann das zu entwickelnde Programm für den Informationsaustausch mit dem Fernwirkerät nur mehr aus einer Reihe von Unterprogrammaufrufen.

Je nach Bedarf können noch weitere Unterprogramme beigefügt werden. Übergeordnete Ausgaben der Netzautomatisierung müssen im allgemeinen den jeweiligen Verhältnissen angepasst werden (Netzkonfigurationen, Leistungsdaten, usw.) [3], da eine so grosse Verallgemeinerung sehr schwierig und vor allem specheraufwendig ist.

Diese Lösung kann bereits bei kleinen Systemen angewendet werden und ermöglicht eine übersichtlich aufgebaute und gut erweiterbare Programmierung. Der Zusatzaufwand, allgemein verwendbare Unterprogramme aus bestehenden Systemen herauszuschälen, zu überarbeiten und zu dokumentieren, ist relativ klein und daher bereits bei einer kleinen Zahl von Anwendungen gerechtfertigt.

## 11. Ausbau des Betriebssystems

Die Erweiterung eines bestehenden Betriebssystems für die Netzautomatisierung ist nur bei grossen Prozessrechnern sinnvoll. In diesem Fall würden aus den Parametern zur Kennzeichnung des Fernwirksystems die Programme zur Abspeicherung der Informationen in die jeweiligen Tabellen im Kernelspeicher automatisch eingebaut werden. Dazu käme wieder die erweiterte Unterprogrammbibliothek für Aufgaben, die nicht völlig allgemein lösbar sind.

Der Entwicklungsaufwand für diese Lösung ist ungleich grösser als für die vorhergehende und nur dann vertretbar, wenn eine entsprechende Zahl von Anwendungen vorliegt oder mit grosser Wahrscheinlichkeit zu erwarten ist. Außerdem muss vorausgesetzt werden können, dass in allen Fällen das gleiche Fernwirksystem und der gleiche Prozessrechner eingesetzt wird, wobei unter «gleich» auch Familien von Fernwirkeräten mit gleichem Aufbau der Informationscodes und Familien von Rechnern, die softwarekompatibel sind, verstanden werden können.

## 12. Die Entwicklung von Software

Wie bereits erwähnt, beschränkt sich die Software-Entwicklung, vor allem bei grösseren Systemen, auf die Anwenderprogramme und den anwendungsgebundenen Teil der Unterprogrammbibliothek (Fig. 2). Je nach Aufgabenstellung ist die Bildung eines kleineren oder grösseren Teams erforderlich, dessen Struktur in Tabelle II angegeben ist [9]. Die Dokumentation wurde hier absichtlich hervorgehoben, da sie die ständige Information aller Mitarbeiter über den letzten Stand der Arbeiten sicherstellt, was für den Erfolg solcher Projekte von entscheidender Bedeutung ist. Die Entwicklung der Software erfolgt etwa in folgenden Schritten:

- Systemanalyse und Festlegung des generellen Lösungsweges;
- Aufteilung in einzelne Aufgaben und Definition der Schnittstellen;
- Organisation der gemeinsamen Datentabellen;
- Entwicklung und Test der Einzelprogramme;
- Prüfung des Zusammenspiels der einzelnen Programme;

*Organisation und Aufteilung der Funktionen innerhalb eines Teams für die Entwicklung eines Programmssystems*

Tabelle II

Leitung (Systemmanagement)	Festlegung des generellen Vorgehens, Zeitplan Entscheidung bei verschiedenen Lösungen Koordination
Dokumentation	Information aller Mitarbeiter Festhalten getroffener Entscheidungen Beschreibung der Lösungen
Peripheriegeräte	Informationsdarstellung (Drucker, Bildsichtgeräte, ...) Befehlseingabe (Tastenfeld, Light Pen, ...)
Schulung	Ausbildung von Mitarbeitern Einführung des Bedienungspersonals
Systemanalyse	Definition der Aufgabenstellung Erarbeiten der Lösungen
Programmierung	Programmentwicklung Einzeltest Systemgenerierung Systemtest

- Systemtest, Simulation des Prozesses;
- Probebetrieb, allfällige Verbesserungen;
- Einschulung des Betriebspersonals;
- Fertigstellung der Dokumentation.

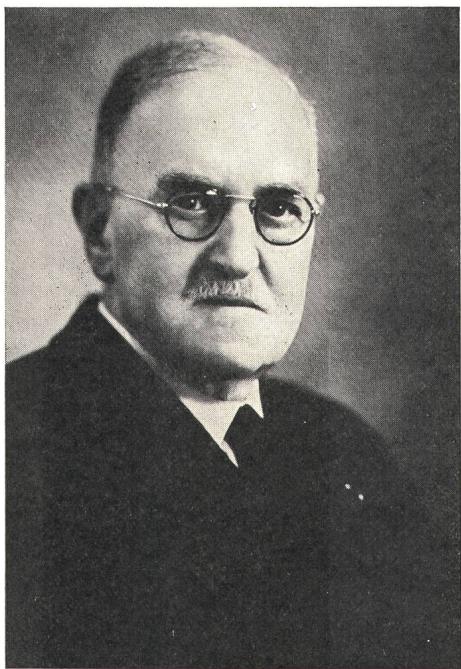
## Literatur

- [1] S. J. Bailey: The control computer-user interface: How to satisfy everybody. Control Engineering 18(1971)5, S. 47...49.
- [2] J. F. Brennan: Computer «reparieren» sich selbst. Entwicklung von Verfahren zur automatischen Fehlersuche und Korrektur. IBM Nachrichten 20(1970)199, S. 7...13.
- [3] P. Nemetz: Load frequency control, economic load dispatch and security in multi-area systems. Centralized Control Systems. IEE Conference Publication 81(1971), p. 9...13.
- [4] J. J. Morris: What to expect when you scale down to a minicomputer. Control Engineering 17(1970)9, p. 65...71.
- [5] N. Frieben und H.-P. Stöckler: Leittechnische Konzeption für Kraftwerke. BBC Sonderheft: Prozessdatenverarbeitung. Mannheim, Brown Boveri & Cie AG, 1971; S. 54...60.
- [6] M. Pohl und H. Zinke: Prozessrechnersystem für Kraftwerke. BBC Sonderheft: Prozessdatenverarbeitung. Mannheim, Brown Boveri & Cie AG, 1971; S. 61...70.
- [7] R. H. Temple and R. E. Daniel: Status of minicomputer software. Control Engineering 17(1970)7, S. 61...64.
- [8] R. Hartwig: Aufbau eines Prozessrechnersystems. IBM Nachrichten 20(1970)199, S. 68...73.
- [9] H. Broekhuis and M. S. Jongkuid: Planning and managing process computer projects. Automation 18(1971)2, p. 60...67.
- [10] P. G. Harhammer: Die Bedeutung von Betriebssystemen bei Prozessrechenanlagen. E und M 88(1971)11, S. 467...471.
- [11] W. Martin: Das Prozess-Datenverarbeitungssystem DP 1000. BBC Sonderheft: Prozessdatenverarbeitung. Mannheim, Brown Boveri & Cie AG, 1971; S. 25...27.
- [12] P. Brunner and W. Werum: Das DP 1000-Betriebssystem. BBC Sonderheft: Prozessdatenverarbeitung. Mannheim, Brown Boveri & Cie AG, 1971, S. 49...52.

## Adresse des Autors:

Dipl.-Ing. Peter Nemetz, Leiter einer Gruppe für die Entwicklung von Applikations-Software auf dem Gebiet der Fernwirktechnik der AG Brown, Boveri & Cie., 5401 Baden.

## Gerard L. F. Philips 1858–1942



Philips AG, Zürich

Der seinerzeit in Zaltbommel am Wal wohnende *Frits Philips* handelte mit Tabak, betrieb eine Kaffeerösterei, besass eine Bank und kaufte das Gaswerk, das er durch dramatische Senkung des Gaspreises zur Rendite brachte. Er hatte 5 Söhne; Gerard der älteste und Anton der jüngste wurden die Leiter der Weltfirma in Eindhoven.

Gerard, am 9. Oktober 1858 geboren, diplomierte 1883 an der Maschinenbauabteilung der Polytechnischen Schule Delft. Schon der an der Oberrealschule in Arnheim wirkende Physiklehrer Dr. H. A. Lorentz hatte in ihm das Interesse für Elektrizität geweckt. In Glasgow, wo er die Installation der elektrischen Lichtanlage auf dem Dampfer «Prins Willem van Oranje» überwachte, erhielt er Gelegenheit, ein Jahr im Laboratorium des Physikers William Thomson (später Lord Kelvin) zu arbeiten. 1887 bestand er am «City and Gilds of London» die Prüfung über elektrische Beleuchtung und Energietransport.

Nach Holland zurückgekehrt übernahm er die Vertretung der AEG, die sich um den Auftrag für das Elektrizitätswerk Rotterdam bewarb. Das Geschäft scheiterte, weil die AEG einen zu hohen Strompreis verlangte. Philips gab 1890 die Stelle auf und sah sich nach einer neuen Tätigkeit um. Bei den Verhandlungen mit Rotterdam hatte er gesehen, wie wichtig es war, gute Glühlampen zu besitzen. Die Fabrikate der vier gerade gegründeten niederländischen Lampenfabriken waren nicht hervorragend. Gerard Philips war überzeugt, Besseres machen zu können. In der Waschküche des Elternhauses richtete er sich ein Laboratorium ein und es gelang ihm rasch, einwandfreie Glühfäden aus Cellulose in gleichbleibender Qualität zu produzieren.

Vater Philips erworb 1891 in Eindhoven eine leerstehende Textilfabrik mit einer Dampfkraftanlage und Gerard ging ans Installieren der Maschinen. Seine Berechnungen hatte er auf eine Tagesproduktion von 1000 Lampen pro Tag basiert. Doch der Ausstoss betrug 1892 nur 11000 und im folgenden Jahr 45000 Stück. Es fehlten gute Verkäufer. Das wurde 1895 schlagartig anders, als Vater Philips seinen jüngsten Sohn Anton dazu bewegen konnte, den Verkauf zu übernehmen.

Gerard Philips, der die technische Leitung und das Laboratorium allein betreute, ent-

wickelte ein neues Verfahren, bei dem der Glasträger mit dem Glühdraht fertig ausgerüstet und als Ganzes in den Glaskolben eingesetzt werden konnte. 1889 konnten die Söhne dem Vater das vorgestreckte Geld zurückzahlen.

Doch es gab neue Probleme. Im Ausland kamen bessere Lampen auf den Markt. Wollte man im Geschäft bleiben, musste man die Neuerungen auch einführen. Gerard Philips gelang nacheinander die Umstellung von Cellulosefäden auf gespritzte, dann auf gezogene Wolframdrähte, alsdann auf gewendete Drähte und gasgefüllte Lampen.

Da die Entwicklungsarbeiten immer umfangreicher wurden, entschloss sich Philips 1911 zur Anstellung von Physikern, Ingenieuren und Chemikern. In wenigen Jahren musste der Fabrikationsapparat viermal umgestellt werden.

Der November 1913 brachte mit einer aufsehenerregenden Beleuchtung der Kalverstraat in Amsterdam mit sog. Halbwattlampen einen grossen Erfolg. Neue Produkte wurden ins Programm aufgenommen: Projektionslampen, Autolampen, Leuchtturmlampen. Während des ersten Weltkrieges, der neben Materialbeschaffungsschwierigkeiten auch Personalmangel brachte, musste Philips Röntgenröhren reparieren, für das holländische Militär Radioröhren bauen; 1919 ging man daran, ganze Radioempfänger zu bauen.

1896 hatte Gerard Philips geheiratet und zum 25. Geschäftsjubiläum (1916) stiftete das Paar einen Fonds, aus dem Kindern der Belegschaft eine höhere Bildung ermöglicht wurde. Im darauffolgenden Jahr verlieh ihm die Universität Delft in Anerkennung seiner Forschungen den Ehrendoktor.

Mitten im zweiten Weltkrieg, am 26. Januar 1942 starb Gerard Philips, nur wenige Tage nach dem Heimgang seiner Gattin. H. Wiiger