



(19)
 Bundesrepublik Deutschland
 Deutsches Patent- und Markenamt

(10) **DE 196 07 515 B4** 2004.04.29

(12)

Patentschrift

(21) Aktenzeichen: **196 07 515.7**
 (22) Anmeldetag: **28.02.1996**
 (43) Offenlegungstag: **12.12.1996**
 (45) Veröffentlichungstag
 der Patenterteilung: **29.04.2004**

(51) Int Cl.7: **G06F 9/46**
G06F 11/30, G06F 15/16

Innerhalb von 3 Monaten nach Veröffentlichung der Erteilung kann Einspruch erhoben werden.

(30) Unionspriorität:
P 7-133920 31.05.1995 JP

(71) Patentinhaber:
Mitsubishi Denki K.K., Tokio/Tokyo, JP

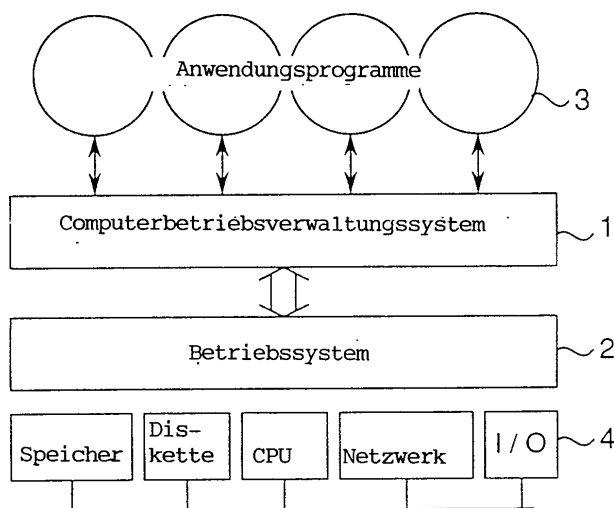
(74) Vertreter:
PRÜFER & PARTNER GbR, 81545 München

(72) Erfinder:
Matsumoto, Hajime, Itami, Hyogo, JP

(56) Für die Beurteilung der Patentfähigkeit in Betracht
 gezogene Druckschriften:
KENAH, Lawrence J., BATE, Simon F.:
VAX/VMS Internals and Data Structures.
Digital Press, 1984, S. 3-12, 183-188,
443-517 u. 704-705;
Heusler, Joachim: Betriebssysteme Siemens
Nixdorf
Informationssysteme AG, 1991, S. 41-58, ISBN:
3-8009-1585-5;
Panagiotis Christinas, UNIX man pages:
top (1), In Man-cgi 1.11, 1994, S. 1-8;
siginfo (5), In Man-cgi 1.11, 1994, S. 1-4;
signal (5), In Man-cgi 1.11, 1994, S. 1-7;
sigaction (2), In Man-cgi 1.11, 1994, S. 1-4;
intro (2), In Man-cgi 1.11, 1994, S. 1, 2, 23-26;

(54) Bezeichnung: **Computer mit Prozessverwalter**

(57) Hauptanspruch: Computer
 – mit einer zentralen Verarbeitungseinheit, einem Hauptspeicher und einem Zusatzspeicher;
 – mit UNIX als Betriebssystem (2), wobei das Betriebssystem (2) den Computer in die Lage versetzt, mehrere Anwendungsprogramme (3) gleichzeitig auszuführen; und
 – mit einem Computerbetriebsverwaltungssystem (1, 11), bestehend aus
 – einem Prozeßverwalter (12), der zwischen dem UNIX-Betriebssystem (2) und den mehreren Anwendungsprogrammen (3) liegt und der
 – einen Ausführungszustandsüberwacher (14) zum Überwachen von Ausführungszuständen der mehreren Anwendungsprogramme (3) dadurch, daß er periodisch eine Nachricht zu jedem Anwendungsprogramm (3) sendet und von den Inhalten der von jedem Anwendungsprogramm (3) zurückgeschickten Nachricht bestimmt, ob das Anwendungsprogramm (3) in Ausführung ist,
 – einen Computerbetriebsmittelverwalter (15) zum Überwachen eines Benutzerstatus der Computerressourcen,
 – einen Fehlerwiederherstellungsprozessor (18) zum Ausführen eines Fehlerwiederherstellungsprozesses, wenn ein Fehler durch den Computerbetriebsmittelverwalter (15) festgestellt worden ist, und
 – einen Endprozessor (17) zum Ausführen eines Nachprozesses, wenn der Ausführungszustandsüberwacher (14)...



Beschreibung

[0001] Die vorliegende Erfindung betrifft einen Computer mit Prozessverwalter, der eine sehr zuverlässige Ausführung und Hochgeschwindigkeitsverarbeitung von einer Mehrzahl von Anwendungsprogrammen, die auf einem einzelnen Computer oder auf einer Mehrzahl von Computern über ein Netzwerk ausgeführt werden, ermöglicht. Insbesondere betrifft sie ein stabiles Betriebsverwaltungssystem für Computer.

[0002] Die der Anmelderin bekannten Computersysteme weisen Hardware auf, wie z.B. den Speicher, Diskettenlaufwerke, eine zentrale Verarbeitungseinheit (CPU) und Netzwerkeinrichtungen, und weisen Betriebssystemsoftware auf, die es ermöglicht, die Hardware effektiv zu nutzen, und weisen Programme auf, die durch den Computer auszuführende Prozeduren speichern, um spezielle Aufgaben auszuführen. Die Person, die den Computer verwendet, führt normalerweise Anwendungsprogramme, die durch den Programmierer in einer Programmiersprache geschrieben sind, durch Anweisen der Betriebssystemsoftware, das Anwendungsprogramm auszuführen, aus oder startet ("run") das gewünschte Anwendungsprogramm durch Mittel. Das Betriebssystem führt dann das eine gewünschte Programm aus, oder es führt gleichzeitig mehrere Programme aus, indem es den Speicher (Hauptspeicher), das Diskettenlaufwerk (Zusatzspeicher), die CPU und die Netzwerkeinrichtungen, die dem Computer zur Verfügung stehen, effektiv nutzt. Einige der Defactostandardbetriebssysteme, die heutzutage verwendet werden, enthalten MVS, verwendet bei Zentralcomputern, VAX, verwendet bei Minicomputern, UNIX, verwendet bei Arbeitsplatzrechnern, und MS-DOS und Windows, verwendet bei Personalcomputern.

[0003] Von diesen bekannten Betriebssystemen ist UNIX in der Lage, gleichzeitig mehrere Anwendungsprogramme auszuführen.

[0004] UNIX wurde ursprünglich durch AT&T in den Vereinigten Staaten von Amerika entwickelt und wird heutzutage hauptsächlich in großem Umfang auf Arbeitsplatzrechnern verwendet. Ursprünglich entwickelt für Softwareentwicklung, computerunterstützter Entwurf (CAD) und andere technische Anwendungen, die von Ingenieuren verwendet werden, bietet UNIX die folgenden Merkmale im Vergleich mit den anderen oben genannten Betriebssystemen.

- (1) UNIX kann gleichzeitig mehrere Programme parallel ausführen.
- (2) Ein Programm, das unter UNIX ausgeführt wird, kann andere Programme dazu bringen, unter UNIX zu laufen und die Anzahl der gleichzeitig laufenden Programme kann sich erhöhen, ohne durch das Betriebssystem begrenzt zu sein.
- (3) Programme, die gleichzeitig ausgeführt werden, werden gleich behandelt und mit der gleichen Priorität ausgeführt, außer es ist für ein oder mehrere Programme festgelegt, mit Vorrang (Pri-

orität) gegenüber einem anderen oder anderen Programmen ausgeführt zu werden. Die Priorität der Ausführung der Programme kann auch dynamisch geändert werden.

(4) Da das virtuelle Speichergebiet eine begrenzte Größe aufweist, verursachen Versuche, ein Programm auszuführen, das diese Grenze überschreitet, daß das Programm sofort beendet wird und daß ein Fehler erzeugt wird (ein "abnormales Ende").

[0005] Die Schwierigkeit mit UNIX ist, daß die Anzahl der Anwendungsprogramme, die gleichzeitig auf einem einzelnen Computer ausgeführt werden können, nicht begrenzt werden kann. Als ein Ergebnis wird, wenn die Anzahl der gleichzeitig ausgeführten Programme die geplante Grenze übersteigt, die Kapazität des Hauptspeichers (Speicher) ungenügend. Wenn der Speicher ungenügend wird, wird zusätzliches virtuelles Speichergebiet auf dem Zusatzspeicher (Diskettenlaufwerk) reserviert, und die Daten im Speicher werden auf die Diskette geschrieben. Wenn jedoch das virtuelle Speichergebiet (virtueller Speicher) für die Programmausführung verwendet wird, wird die Ausführungsgeschwindigkeit des Programmes um einige 10% bis einige 100% fallen. Bei einigen Anwendungsprogrammen ist es nur umständlich, wenn die Ausführungsgeschwindigkeit fällt, aber bei Systemen, die Echtzeitausführung verlangen, kann die reduzierte Ausführungsgeschwindigkeit eine Quelle für Schwierigkeiten werden.

[0006] Wenn beide, der Hardwarespeicher und der virtuelle Speicher, ungenügend sind, wird UNIX eines der Anwendungsprogramme zwangsweise beenden. Da es nicht möglich ist, die Reihenfolge (Priorität) festzulegen, in der die Programme zwangsweise beendet werden, werden Programme, für die Zuverlässigkeit wesentlich ist, und Programme, für die Zuverlässigkeit nicht wesentlich ist, zwangsweise mit derselben Wahrscheinlichkeit beendet. Wenn ein Programm zwangsweise beendet wird, ist die Funktionalität des Systems auch verloren.

[0007] UNIX weist auch keine Mittel auf zum zentralen Verwalten von Änderungen, die in einem Programm aufgetreten sind, wann diese Änderungen aufgetreten sind, von den Fehlermeldungen, die erzeugt wurden, als das Programm gestartet, normal beendet oder abnormal beendet wurde. Als ein Ergebnis davon gibt es keine Möglichkeit, zu bestimmen, welches Programm die Ursache der Schwierigkeit war, wenn sich ein Systemfehler oder eine Schwierigkeit entwickelt. Während einige Fehlermeldungen ausgegeben werden, sind die Meldungen nicht durch die Maschine (CPU), durch die Ausgabevorrichtung oder durch das Ausgabeverfahren vereinheitlicht, und eine Fehlersuche ist daher schwierig und zeitaufwendig.

[0008] UNIX behandelt jedes der Mehrzahl von gleichzeitig ausgeführten Programmen gleich und ist nicht in der Lage, die Ausführungszeit (CPU-Zeit), die

jedem Programm zugeordnet ist, zu steuern. Als ein Ergebnis verringern Programme, die ungewöhnlich große Anforderungen an die CPU und den Speicher stellen, die gesamte Antwortzeit und den Durchsatz des Systems.

[0009] Bei großen Systemen, die auf UNIX basierende Computer über ein gemeinsames Netzwerk miteinander verbinden, können die oben beschriebenen Schwierigkeiten häufig auftreten und das Auffinden der Quelle der Schwierigkeiten ist noch schwieriger. Es gibt auch keine Mittel zum sofortigen Benachrichtigen des Bedieners, daß eine der oben aufgeführten Schwierigkeiten aufgetreten ist, und die Wiederherstellungsanstrengungen sind daher verzögert.

Stand der Technik

[0010] Aus KENAH, Lawrence J.; BATE, Simon F.: VAX/VMS Internals and Data Structures, Digital Press, 1984, ist bekannt, daß das VAX/VMS-Betriebssystem in der Lage ist, mehrere Anwendungsprogramme gleichzeitig auszuführen und deren Ausführungsstatus zu überwachen.

[0011] Aus den UNIX main pages von Panagiotis Christinas vom 11. Januar 1994 können Einzelheiten des Betriebssystems UNIX entnommen werden.

Aufgabenstellung

[0012] Es ist Aufgabe der vorliegenden Erfindung, einen Computer vorzusehen, der mehrere Anwendungsprogramme gleichzeitig, stabil und zuverlässig ausführen kann.

[0013] Die Aufgabe wird durch den Computer des Anspruchs 1 gelöst.

[0014] Weiterbildungen der Erfindung sind in den Unteransprüchen angegeben.

[0015] Die Erfindung umfaßt auch ein Computersystem nach Anspruch 9.

[0016] Nachfolgend werden die Lösung der Aufgabe und Vorteile der Lösung anhand von als Aspekten bezeichneten Ausführungsformen detailliert beschrieben.

[0017] Durch das Computersystem wird eine stabile und sehr zuverlässige Ausführung von individuellen Anwendungsprogrammen durch Begrenzen der Anzahl der gleichzeitig ausführbaren Anwendungsprogramme und Überwachung der Anwendungsprogrammausführung, der Computerelemente und des Netzwerkes möglich.

[0018] Der Prozeßverwalter ist zwischen den gleichzeitig ausgeführten Anwenderprogrammen und dem Betriebssystem angeordnet. Das Betriebssystem ist dadurch charakterisiert, daß es in der Lage ist, gleichzeitig mehrere Anwendungsprogramme auszuführen, von einem ausgeführten Programm ein neues darauf bezogenes Programm zu erzeugen, mit dem kommuniziert werden kann oder das direkt verwaltet werden kann, alle ausgeführten Programme gleich zu behandeln, außer das Betriebssystem ist

während dem Programmstart speziell anders benachrichtigt worden, dynamisch die Prioritätssequenz der ausgeführten Programme zu ändern. Es weist einen virtuellen Speicherbereich von endlicher Größe auf, und ist in der Lage, die Programmausführung sofort abnormal zu beenden, wenn das auszuführende Programm die Grenzen des virtuellen Speicherbereiches überschreitet. Das Computerbetriebsverwaltungssystem weist weiter eine Schnittstelle zu beiden auf, dem Betriebssystem und den individuellen Anwendungsprogrammen, und tritt dem Betriebssystem gegenüber als einzelnes Anwendungsprogramm auf. Die Programmdefinitionsdatei ist durch den Anwender vordefiniert, um die maximale Anzahl der gleichzeitig ausführbaren Programme, die Ausführungspriorität des Programms, das Fehlerwiederherstellungsverfahren, Programme, die durch Kommunikation mit anderen Programmen auf dem Netzwerk ausgeführt werden und Informationen über das nach der Programmbeendigung für jede Programmkategorie auszuführende Nachverfahren zu speichern. Eine Warteschlangeprogrammaufnahmedatei bzw. eine Vormerkprogrammaufnahmedatei speichert die Informationen der vorgemerkten Programme, die darauf warten, ausgeführt zu werden, und eine Protokolldatei speichert Protokolldaten, wenn ein Programm startet oder endet oder wenn ein Fehler auftritt.

Ausführungsbeispiel

[0019] Es folgt die Beschreibung von Ausführungsbeispielen anhand der Figuren. Von den Figuren zeigen

[0020] **Fig. 1** ein Konzeptdiagramm der Beziehung zwischen dem Betriebssystem, den Anwendungsprogrammen und dem Computerbetriebsverwaltungssystem;

[0021] **Fig. 2** ein Blockdiagramm des Computerbetriebsverwaltungssystems;

[0022] **Fig. 3** ein Blockdiagramm und ein Ablaufdiagramm der verschiedenen Steuereinheiten einer Ausführungsform des Computerbetriebsverwaltungssystems;

[0023] **Fig. 4** ein Flußdiagramm des Betriebes der Ausführungssteuereinheit einer Ausführungsform eines Computerbetriebsverwaltungssystems;

[0024] **Fig. 5** ein Flußdiagramm des Betriebes des Ausführungszustandsüberwachers einer Ausführungsform eines Computerbetriebsverwaltungssystems;

[0025] **Fig. 6** ein Flußdiagramm des Betriebes der Ausführungsprioritätssteuereinheit, wenn ein Programmstart in dem Computerbetriebsverwaltungssystem angefordert wird;

[0026] **Fig. 7** ein Flußdiagramm des Betriebes der Ausführungsprioritätssteuereinheit, wenn die Prioritätsreihenfolge der auszuführenden Programme in dem Computerbetriebsverwaltungssystem dynamisch geändert wird;

[0027] **Fig. 8** ein Flußdiagramm mit Betrieb des Fehlerwiederherstellungsprozessors in dem Computerbetriebsverwaltungssystem;

[0028] **Fig. 9** eine Tabelle eines Beispiels von Verfahrensdefinitionen für den Fehlerwiederherstellungsprozeß in dem Computerbetriebsverwaltungssystem;

[0029] **Fig. 10** ein Flußdiagramm des Betriebes des Endprozessors in dem Computerbetriebsverwaltungssystem;

[0030] **Fig. 11** ein Flußdiagramm des Betriebes des Computer-Betriebsmittelverwalters in dem Computerbetriebsverwaltungssystem;

[0031] **Fig. 12** ein Flußdiagramm des Betriebes des Netzwerkverwalters in dem Computerbetriebsverwaltungssystem;

[0032] **Fig. 13** ein Flußdiagramm des Betriebes des Fehlerkommunikationsmittels, das die Fehlerkommunikationsfunktion des Computerbetriebsverwaltungssystems realisiert;

[0033] **Fig. 14** ein Flußdiagramm des Betriebes des Fehlerzeichengebers, der die Fehlerkommunikationsfunktion und die Fehlerüberwachungsfunktion des Computerbetriebsverwaltungssystems realisiert;

[0034] **Fig. 15** ein Flußdiagramm und ein Blockdiagramm von verschiedenen Steuereinheiten einer zweiten Ausführungsform eines Computerbetriebsverwaltungssystems.

[0035] Im folgenden werden Ausführungsformen eines Computerbetriebsverwaltungssystems entsprechend mit Bezug zu den Figuren beschrieben.

Ausführungsform 1:

[0036] **Fig. 1** ist ein Konzeptdiagramm des Betriebsverwaltungssystems in einem Computer. Wie in **Fig. 1** gezeigt ist, ist das Computerbetriebsverwaltungssystem **1** zwischen dem Betriebssystem **2** und den Anwendungs-/Applikationsprogrammen **3** angeordnet. Das Computerbetriebsverwaltungssystem **1** arbeitet, von der Perspektive des Betriebssystems **2** aus gesehen, als ein einzelnes Anwendungsprogramm, aber es weist eine Schnittstelle zu den Applikationsprogrammen **3** auf, wodurch das Computerbetriebsverwaltungssystem **1** aus Sicht der Anwendungsprogramme **3** als Teil des Betriebssystems **2** erscheint. Die Funktion des Computerbetriebsverwaltungssystems **1** ist es, die Ausführung der Anwendungsprogramme **3** zu überwachen und zu steuern.

[0037] Das Betriebssystem **2** weist die folgenden Merkmale auf.

- (1) Die Fähigkeit, mehrere Anwendungsprogramme parallel auszuführen.
- (2) Die Fähigkeit, ein neues Programm (Tochterprogramm) von einem ausgeführten Programm (Elternprogramm) zu erzeugen. Die Anzahl der gleichzeitig ausgeführten Programme erhöht sich, ohne durch das Betriebssystem begrenzt zu sein.
- (3) Gleichzeitig ausgeführte Programme werden mit der gleichen Priorität ausgeführt und behan-

delt, außer es ist etwas anderes festgelegt. Die Priorität der ausgeführten Programme kann festgelegt werden und dynamisch geändert werden.

(4) Der virtuelle Speicherbereich weist eine endliche Größe auf. Wenn ein Programm ausgeführt wird, das diese endliche Größe überschreitet, wird das Programm sofort beendet und ein abnormales Ende (Fehler) wird erzeugt.

[0038] Der Aufbau des in **Fig. 1** gezeigten Computerbetriebsverwaltungssystems **1** wird in **Fig. 2** gezeigt und im folgenden beschrieben.

[0039] Wie in **Fig. 2** gezeigt ist, enthält das Computerbetriebsverwaltungssystem **1** einen Prozeßverwalter **12**, eine Programmdefinitionsdatei **21**, eine Aufnahme-datei für vorgemerkte Programme (Anforderungswarteschlange) **22** und eine Protokolldatei **23**. Der Prozeßverwalter **12** ist ein Programm, das unter dem Betriebssystem arbeitet und das entweder im Hauptspeicher (Speicher) oder in einer ergänzenden Speichereinrichtung (Diskettenlaufwerk) des Computers angeordnet ist, um das Ausführen der Programme zu überwachen und zu steuern. Die Programmdefinitionsdatei **21** ist in der ergänzenden Speichereinrichtung (Diskettenlaufwerk) vorgesehen und ist das Mittel zum Aufzeichnen von Informationen über die ausgeführten Programme und enthält die maximale Anzahl der gleichzeitig ausführbaren Programme für jede Programmkategorie, die Ausführungspriorität von jedem Programm, die Prozedur, die auszuführen ist, wenn ein Fehler auftritt, und die Grenzen der Computerressourcen. Die Anforderungswarteschlangendatei **22** ist auch in der ergänzenden Speichereinrichtung (Diskettenlaufwerk) vorgesehen und ist das Mittel zum Aufnehmen von Informationen über Programme, für die der Start der Ausführung während des Programmstarts eingeschränkt ist. Die Protokolldatei **23** ist auch in der ergänzenden Speichervorrichtung (Diskettenlaufwerk) vorgesehen und ist das Mittel zum Aufnehmen der Protokollinformationen, die den Programmstart, das Programmende und Programmfehler betreffen.

[0040] Wie in **Fig. 2** gezeigt ist, enthält die Programmdefinitionsdatei **21** die folgenden Daten: die maximale Anzahl der gleichzeitig ausführbaren Programme für jede Programmkategorie, die Ausführungspriorität für jedes Programm, die auszuführenden Prozeduren, wenn ein Fehler während der Programmausführung auftritt, und die auszuführenden Prozeduren, wenn ein Programm normal endet. Alle diese Datenparameter werden durch den Benutzer definiert. Die Programmkategorien sind genauer durch den Benutzer vordefinierte Gruppen von Programmen und zeigen den Typ des Programms an. Die maximale Anzahl der gleichzeitig ausführbaren Programme, die Prioritätssequenz, die Fehlerwiederherstellungsprozeduren und die Endprozedur sind für jede Programmkategorie festgelegt. Auf die in der Programmdefinitionsdatei gespeicherten Daten wird beim Programmstart, beim Programmende und,

wenn ein Fehler auftritt, Bezug genommen, um die definierten Prozesse auszuführen.

[0041] Die Aufnahme­datei für vorgemerkte Programme (Anforderungswarteschlange) **22** ist ein Pufferbereich zum Halten der Programme, die aufgrund der Ausführungsgrenzen warten, ausgeführt zu werden. Die Programme sind in der Aufnahme­datei für vorgemerkte Programme **22** in absteigender Reihenfolge der Programmausführungspriorität für jede Programmkategorie gespeichert. Genauer gibt es mehrere Anforderungswarteschlangen für jede Programmkategorie in der Aufnahme­datei für vorgemerkte Programme **22**, und die vorgemerkten Programme sind innerhalb jeder Anforderungswarteschlange in absteigender Reihenfolge von dem Programm mit der höchsten Ausführungspriorität gespeichert. Als ein Ergebnis werden die Programme beginnend von dem Programm mit der höchsten Priorität ausgeführt.

[0042] Bei einem Programmstart speichert die Protokolldatei **23** ein Startprotokoll, das den Namen des gestarteten Programms, die Zeit und das Datum, an dem das Programm gestartet wurde, den Computernamen und die Programmnummer beim Start enthält. Wenn ein Fehler auftritt, speichert die Protokolldatei **23** ein Fehlerprotokoll, das den Programmnamen, die Zeit und das Datum, an dem der Fehler auftrat, den Computernamen, die Programmnummer und den Fehlerstatus enthält. Der Fehlerstatus ist die Abschlußnachricht des Betriebssystems oder die Bedingungen, als der Fehler auftrat, und wird durch den Prozeßverwalter **12** zur Verfügung gestellt. Wenn ein Programm endet, speichert die Protokolldatei **23** ein Endprotokoll, das den Programmnamen, die Zeit und das Datum, an dem das Programm endete, den Computernamen, die Programmnummer und den Endstatus (normales oder abnormales Ende) enthält. Auf die gespeicherten Protokolldateien kann durch den Benutzer zugegriffen werden, um den Betriebsstatus des Systems, für die Fehlersuche den Grund eines Fehlers festzustellen, oder falls es anderweitig nötig ist.

[0043] Der in **Fig. 2** gezeigte Prozeßverwalter **12** weist auch die folgenden Steuereinheiten **13** – **20** auf.

[0044] Die Ausführungsstee­ereinheit **13** steuert die Ausführung der einzelnen Anwendungsprogramme entsprechend den Anwendungsprogrammstartanweisungen, die entweder von dem Bediener oder von dem Programm empfangen werden.

[0045] Das Ausführungsstatusüberwachungsmittel/Zustandsüberwacher **14** überwacht den Ausführungsstatus/-zustand der Anwendungsprogramme durch Austausch von Nachrichten mit den ausgeführten Anwendungsprogrammen. Wenn in einem der ausgeführten Programme ein Fehler auftritt, benachrichtigt das Ausführungsstatusüberwachungsmittel **14** die Steuereinheit, die den vordefinierten Fehlerprozeß ausführt.

[0046] Der Computer-Ressourcen-Verwalter/-Betriebsmittelverwalter **15** überwacht die Verwendung

der Computerressourcen einschließlich der CPU, des Speichers und des Diskettenlaufwerks durch Vergleichen der von der Programmdefinitionsdatei **21** gelesenen Grenzwerte mit der aktuellen Benutzung. Wenn ein Fehler auftritt, benachrichtigt der Computer-Ressourcen-Verwalter **15** den Fehlerwiederherstellungsprozessor, der den vordefinierten Fehlerwiederherstellungsprozeß ausführt, und das Fehlerkommunikationsmittel, das den vordefinierten Fehlerbenachrichtigungsprozeß ausführt.

[0047] Die Ausführungsprioritätsstee­ereinheit **16** steuert die Prioritätsreihenfolge der Programme in der Aufnahme­datei für vorgemerkte Programme (Anforderungswarteschlange) **22** und die Prioritätsreihenfolge der ausgeführten Programme entsprechend der von der Programmdefinitionsdatei **21** gelesenen Programmprioritätssequenz.

[0048] Der Endprozessor **17** führt den Nachprozeß (der Prozeß, der ausgeführt wird, nachdem ein Programm endet) entsprechend der in der Programmdefinitionsdatei **21** gespeicherten Nachprozeßprozedur aus, wenn ein Programm endet, und speichert das Endprotokoll in der Protokolldatei **23**.

[0049] Der Fehlerwiederherstellungsprozessor **18** führt den Fehlerwiederherstellungsprozeß entsprechend dem in der Programmdefinitionsdatei **21** gespeicherten vordefinierten Fehlerwiederherstellungsprozeß aus, wenn ein Fehler auftritt, und speichert ein Fehlerprotokoll in der Protokolldatei **23**.

[0050] Das Fehlerkommunikationsmittel/der Fehlerzeichengeber **19** informiert den Bediener unmittelbar nach dem Auftreten eines Fehlers mittels beispielsweise eines hörbaren Alarms oder anderer Kommunikationsmittel entsprechend dem in der Programmdefinitionsdatei **21** gespeicherten Fehlerwiederherstellungsprozeß.

[0051] Der Netzwerkverwalter **20** kommuniziert mit den Prozeßverwaltern **12**, die auf anderen Computern des Kommunikationsnetzwerks ausgeführt werden, um Protokollinformationen betreffend den Programmstart, Fehlern und Programmende auszutauschen. Der Prozeßverwalter **12** von einem Computer kann damit den Ausführungsstatus der Programme auf anderen Computern des Netzwerks festlegen bzw. bestimmen und eine zentralisierte Betriebsverwaltung kann mittels eines einzelnen Computers erreicht werden.

[0052] **Fig. 3** ist ein Blockdiagramm und ein Flußdiagramm der verschiedenen Steuereinheiten einer Ausführungsform des Computerbetriebsverwaltungssystems **1** entsprechend der vorliegenden Erfindung.

[0053] Wenn ein Anwendungsprogramm ausgeführt werden soll, gelangt eine Ausführungsanforderung an das Computerbetriebsverwaltungssystem **1** und nicht an das Betriebssystem. Wenn eine Programmstartanforderung durch den Bediener oder von einem anderen Programm an das Computerbetriebsverwaltungssystem **1** (S21) gegeben wird, liest die Ausführungsstee­ereinheit **13** die maximale Anzahl der

gleichzeitig ausführbaren Programme von der Programmdefinitionsdatei **31** (S22) und vergleicht diese maximale Anzahl mit der Anzahl der ausgeführten Programme. Wenn die Anzahl der ausgeführten Programme kleiner als die gelesene Grenze (maximale Anzahl) ist, wird eine Tochterprogrammstartanforderung an das Betriebssystem (S23) gegeben. Wenn die Anzahl der ausgeführten Programme mit der Grenze (maximale Anzahl) übereinstimmt und eine Programmstartanforderung herausgegeben wird, wird das angeforderte Programm in der Aufnahme-datei für vorgemerkte Programme **33** vorgemerkt.

[0054] Die Ausführungssteuereinheit **13** überwacht regelmäßig die Anzahl der ausgeführten Programme. Als ein Ergebnis wird, wenn es möglich wird ein vorgemerktes Programm auszuführen, das nächste auszuführende Programm von der Aufnahme-datei für vorgemerkte Programme **33** gelesen und eine Startanforderung wird an das Betriebssystem (S23) gegeben. Die Ausführungssteuereinheit **13** speichert dann das Startprotokoll in der Protokoll-datei **32**, wenn der Programmstart endet.

[0055] Die so ausgeführten Anwendungsprogramme **3** werden so ausgeführt, daß eine Eltern-Tochterbeziehung mit dem Prozeßverwalter **12** erhalten wird, so daß Informationen mit dem Prozeßverwalter **12** in dem Computerbetriebsverwaltungssystem **1** immer ausgetauscht (kommuniziert) werden können. Wenn ein neues Programm ausgeführt wird, erkennt das Betriebssystem **2** das Programm (d.h. der Prozeßverwalter **12** in dem Computerbetriebsverwaltungssystem **1**) und führt das neue Programm als "Eltern" und das neue Programm als "Tochter" aus. Das Mittel zum Durchführen dieser Eltern-Tochterkommunikation ist dadurch gekennzeichnet, daß es deutlich einfacher ist als das Mittel, das für eine Kommunikation zwischen Programmen mit unterschiedlicher Beziehung benötigt wird. Die Anwendungsprogramme, die durch den Prozeßverwalter **12** verwaltet werden, sind relativ zu dem Prozeßverwalter **12** immer "Tochter" und sind so aufgebaut und verwaltet, daß "Tochter der Tochter", d.h. "Enkel"-Programme nicht auftreten.

[0056] Wenn ein Programm einmal gestartet ist, überwacht das Ausführungsstatusüberwachungsmittel **14** den Ausführungsstatus von jedem ausgeführten Programm durch Senden von Nachrichten zu und Empfangen von Antworten von jedem ausgeführten Programm (S24). Wenn ein Fehler mit einem überwachten Programm auftritt, werden der Fehlerwiederherstellungsprozessor **18** und das Fehlerkommunikationsmittel **19** entsprechend informiert, um die vordefinierten Prozesse (S29 und S30) auszuführen.

[0057] Wenn es eine Anforderung von dem Bediener oder einem anderen Programm während der Programmausführung gibt, die Prioritätssequenz eines ausgeführten Programmes oder eines vorgemerkten Programmes in der Aufnahme-datei für vorgemerkte Programme (Anforderungswarteschlange) **33** zu ändern, ändert die Ausführungsprioritätssequenzsteu-

ereinheit **16** die Prioritätssequenz der ausgeführten Programme oder der Programme in der Aufnahme-datei für vorgemerkte Programme **33** (S28).

[0058] Gleichzeitig zu der Überwachung der Programmausführung durch das Ausführungsstatusüberwachungsmittel **14** überwacht der Computer-Ressourcen-Verwalter **15** die Computerressourcen (S25) durch Vergleichen der aktuellen Benutzung der Computerressourcen (einschließlich der CPU-Zeit, Speicher und Diskettenlaufwerk) mit den in der Programmdefinitionsdatei definierten Ressourcenverwendungsgrenzen. Wenn ein Fehler auftritt, werden der Fehlerwiederherstellungsprozessor **18** und das Fehlerkommunikationsmittel **19** benachrichtigt, um die vordefinierten Prozesse auszuführen (S29 und S30).

[0059] Als Teil der Ausführungsstatusüberwachung und der Computerressourcenverwaltungsroutinen werden Programme, die während der Kommunikation mit anderen Programmen auf dem Netzwerk ausgeführt werden, auch mittels des Netzwerkverwalters **20** überwacht durch Austausch von Informationen mit den Prozeßverwaltern der anderen Computer **34** auf dem Netzwerk (**31**).

[0060] Die Statusausführungsüberwachung (S24) und die Computerressourcenverwaltung (S25) werden dann in einer andauernden Schleife ausgeführt bis das Betriebssystem dem Prozeßverwalter mitteilt, daß ein Programm beendet wurde.

[0061] Wenn ein Fehler während der Ausführungsstatusüberwachung oder der Computerressourcenverwaltung auftritt, wird die Fehlerinformation dem Fehlerwiederherstellungsprozessor **18** zur Verfügung gestellt, der den Wiederherstellungsprozeß entsprechend der in der Programmdefinitionsdatei gespeicherten Wiederherstellungsprozedur ausführt und der gleichzeitig das Fehlerprotokoll in der Protokoll-datei **32** speichert (S29). Das Fehlerkommunikationsmittel **19** gibt auch einen entsprechenden Alarm heraus oder eine andere Benachrichtigung entsprechend dem in der Programmdefinitionsdatei **21** gespeicherten Fehlerwiederherstellungsprozeß (S30).

[0062] Wenn der Prozeßverwalter durch das Betriebssystem benachrichtigt wird, daß ein Programm beendet wurde, führt der Endprozessor **17** den in der Programmdefinitionsdatei gespeicherten Endprozeß aus und schreibt das Endprotokoll in die Protokoll-datei (S27).

[0063] Die allgemeine Funktion der verschiedenen Steuereinheiten des Computerbetriebsverwaltungssystem **1** entsprechend der vorliegenden Erfindung wurden oben beschrieben. Eine detailliertere Beschreibung ihres Betriebes wird im folgenden beschrieben.

[0064] **Fig. 4** ist ein Flußdiagramm des Betriebes der Ausführungssteuereinheit **13** des Prozessverwalters **12** in dem Computerbetriebsverwaltungssystem **1** entsprechend der vorliegenden Erfindung.

[0065] Programmstartanforderungen, die von dem Bediener oder von einem anderen Programm zu dem

Computerbetriebsverwaltungssystem **1** gegeben werden (S41), werden in dem Prozeßverwalter **12** eingegeben, der dann die maximale Anzahl der Programmkopien von der Programmdefinitionsdatei **21** liest, die gleichzeitig ausgeführt werden können (S42). Die maximale Anzahl der gleichzeitig ausführbaren Programme kann (1) die gesamte Anzahl der Programme sein, die gleichzeitig auf dieser Maschine ausgeführt werden können, oder kann (2) die Anzahl der Kopien von einer Programmkategorie sein, die gleichzeitig ausgeführt werden können. Die Anzahl der Programme des gleichen Typs, die momentan auf dem Computer (Maschine) laufen, wird auch gezählt. Die von der Programmdefinitionsdatei **21** gelesene Steuergrenze wird mit der Anzahl der ausgeführten Programme verglichen (S44). Wenn die Anzahl der ausgeführten Programme kleiner ist als die Steuergrenze, kann das Programm sofort ausgeführt werden und der Programmstart wird fortgesetzt (S45). Die Startprozedur weist das Betriebssystem an, daß das Programm als "Tochter" (Tochterprozeß) des Prozeßverwalters gestartet wurde. Wenn der normale Start vollendet ist, werden der Name des gestarteten Programms, die Zeit und das Datum, der Computernamen und die Programmnummer beim Start in der Protokolldatei **23** als das Startprotokoll gespeichert (S46).

[0066] Wenn während der Programmausführung die Anzahl der momentan ausgeführten Programme kleiner ist als die Steuergrenze, wird das Programm in die Aufnahmedatei für vorgemerkte Programme (Anforderungswarteschlange) **22**, der Pufferbereich für Programme, die darauf warten ausgeführt zu werden, geschrieben.

[0067] Der Prozessverwalter **12** überprüft regelmäßig ob vorgemerkte Programme in der Aufnahmedatei für vorgemerkte Programme (Anforderungswarteschlange) **22**, für die eine Ausführungsanforderung ausgegeben wurde, ausgeführt werden können. Dieser Ausführungsanfrageprozeß wird fortgeführt, um festzustellen ob vorgemerkte Programme ausgeführt werden können soweit die Anzahl der momentan ausgeführten Programme die Steuergrenze überschreitet. Wenn die Anzahl der momentan ausgeführten Programme kleiner wird als die Steuergrenze, wird das Betriebssystem instruiert mit der Programmausführung fortzufahren (S45).

[0068] **Fig. 5** ist ein Flußdiagramm des Betriebes des Ausführungsstatusüberwachungsmittels **14** in einer Ausführungsform des Computerbetriebsverwaltungssystems entsprechend der vorliegenden Erfindung.

[0069] Wie in **Fig. 5** gezeigt ist, startet der Prozeßverwalter **12** ein Anwendungsprogramm als "Tochter" des Prozeßverwalters **12**, basierend auf der Programmstartanforderung (S51). Das Betriebssystem verfolgt das Programm (Prozeß), das den Start eines Programms verwaltet, als den "Eltern-Prozeß" und das Programm, das somit als "Tochter-Prozeß" gestartet wird. Das Mittel zur Kommunikation zwischen

den Programmen in einer Eltern-Tochterbeziehung ist einfach.

[0070] Alle Anwendungsprogramme werden als "Tochter" des Prozeßverwalters **12** ausgeführt. Durch eine regelmäßige Kommunikation mit den ausgeführten Programmen ist der Prozeßverwalter **12** in der Lage den Ausführungsstatus von jedem Anwendungsprogramm zentral zu verwalten.

[0071] Wie wiederum in **Fig. 5** gezeigt ist, sendet das Ausführungsstatusüberwachungsmittel **14** zu jedem Anwendungsprogramm unter seiner Verwaltung in regelmäßigen Zeitabständen eine Nachricht, um festzustellen ob jedes abgefragte Programm noch ausgeführt wird (S52).

[0072] Das abgefragte Anwendungsprogramm sendet dem Ausführungsstatusüberwachungsmittel **14** eine Antwortnachricht, die bestimmt auf welche Bestätigungsnachricht das Anwendungsprogramm antwortet. Der Prozeßverwalter **12** bestätigt daß das abgefragte Anwendungsprogramm noch ausgeführt wird indem er die Antwort innerhalb eines vorbestimmten Zeitraums empfängt (S53). Der Prozeßverwalter **12** überwacht den Ausführungsstatus von jedem Programm durch Wiederholen dieses Nachrichtenaustauschs mit jedem Anwendungsprogramm bis das Betriebssystem dem Prozeßverwalter **12** mitteilt, daß ein spezielles Anwendungsprogramm beendet ist (S54).

[0073] Alle Anwendungsprogramme sind so geschrieben daß sie sofort antworten, wenn ein Bestätigungsnachricht von dem Prozeßverwalter **12** empfangen wird. Diese Funktion kann dadurch erreicht werden daß die benötigten Unterroutinen als Bibliothek vorgesehen werden, die eingefügt werden wenn das Programm geschrieben und übersetzt wird.

[0074] Wenn die Antwort nicht innerhalb der vorbestimmten Zeitperiode empfangen wird (S53), bestimmt der Prozeßverwalter **12** das ein Fehler in dem Anwendungsprogramm aufgetreten ist, und benachrichtigt den Fehlerwiederherstellungsprozessor **18** und das Fehlerkommunikationsmittel **19** (S55, S56). Der Fehlerwiederherstellungsprozessor **18** und das Fehlerkommunikationsmittel **19** führen dann entsprechend vorbestimmte Prozeduren aus.

[0075] Es ist dadurch für das Computerbetriebsverwaltungssystem möglich, die Fehler festzustellen, wenn ein Fehler in einem ausgeführten Programm auftritt, und es kann die entsprechende Fehlerwiederherstellung oder den Umschaltprozeß ausführen. Fehler, die durch ein Programm erzeugt werden, können oft weder durch das Programm selbst noch durch das Betriebssystem festgestellt werden. Dies macht ein durch die vorliegende Erfindung beschriebenes Computerbetriebsverwaltungssystem notwendig für den Betrieb von Systemen, die eine hohe Zuverlässigkeit erfordern.

[0076] Ein Verfahren zum gleichzeitigen Ausführen von mehreren Programmen mittels der Ausführungssteuereinheit **13** und des Ausführungsstatusüberwachungsmittels **14** des Computerbetriebsverwaltungs-

systems entsprechend der vorliegenden Erfindung wird im folgenden beschrieben.

[0077] Wenn eine neue Programmstartanforderung entweder durch den Bediener oder durch ein Programm ausgegeben wird, startet der Prozeßverwalter **12**, wie oben beschrieben, das Programm als Tochter-Prozeß. Nach dem Programmstart überwacht das Ausführungsstatusüberwachungsmittel **14** den Ausführungsstatus des Programmes. Von diesem Zustand an ist es für den Prozeßverwalter **12** möglich eine andere neue Programmstartanforderung zu empfangen. Wenn dann eine andere Programmstartanforderung empfangen wird, wird das neue Programm durch die Ausführungssteuereinheit **13** gestartet und der Ausführungsstatus wird dann durch das Ausführungsstatusüberwachungsmittel **14** überwacht. Zu diesem Zeitpunkt überwacht der Prozeßverwalter **12** die Ausführung von zwei Programmen und kann eine andere Programmstartanforderung empfangen. Diese einfache Schleife wird danach wiederholt und somit wird es möglich, die Anzahl der gleichzeitig ausgeführten Programme zu erhöhen.

[0078] **Fig. 6** ist ein Flußdiagramm eines Betriebes der Ausführungsprioritätssequenzsteuereinheit **16**, wenn in dem Computerbetriebsverwaltungssystem nach der vorliegenden Erfindung ein Programmstart gefordert wird.

[0079] Wie in **Fig. 6** gezeigt ist, liest die Ausführungsprioritätssequenzsteuereinheit **16** die Prioritätsreihenfolge des Programms und die maximale Anzahl der gleichzeitig ausführbaren Programme, die durch die Programmgruppe verwaltet werden, von der Programmdefinitionsdatei **21** (S62), wenn eine Programmstartanforderung in dem Prozeßverwalter **12** (S61) eingegeben wird. Die Ausführungsprioritätssequenzsteuereinheit **16** bestätigt dann bzw. stellt dann die Anzahl der momentan ausgeführten Programme und die Anzahl der Programme der gleichen Programmkategorie (S63) fest.

[0080] Wenn die Anzahl der momentan ausgeführten Programme kleiner ist als die Anzahl der ausführbaren Programme, die von der Programmdefinitionsdatei ausgelesen wurde, wird das Programm als Tochterprogramm des Prozeßverwalters gestartet durch Mitteilen der Prioritätsreihenfolge des Programmes an das Betriebssystem und durch Anweisen des Betriebssystems, das Programm zu starten. Wenn versucht wird ein Programm zu starten, das die Steuergrenze der Anzahl der gleichzeitig ausführbaren Programme überschreitet, wird die Ausführung sofort unterbrochen, wird ein definierte Prioritätsreihenfolge (Initialisierungswert) an das Programm angehängt und wird das Programm vorübergehend in der Aufnahme-datei für vorgemerkte Programme (Anforderungswarteschlange) **22** als eine Ausführungsanforderung (S66) gespeichert. Die Prioritätsreihenfolge des vorgemerkten Programms wird mit der der anderen Programme in der Anforderungswarteschlange **22** verglichen und die Anforderungswar-

teschlange (**22**) wird in absteigender Reihenfolge der Priorität sortiert (S67).

[0081] Der Prozeßverwalter **12** überwacht regelmäßig die Anzahl der ausgeführten Programme, so daß das Programm an der Spitze der Warteschlange in den Speicher gebracht wird und durch die Ausführungssteuereinheit ausgeführt wird, wenn die Ausführung eines der Programme in der Anforderungswarteschlange **22** möglich wird.

[0082] **Fig. 7** ist ein Flußdiagramm des Betriebes der Ausführungsprioritätssequenzsteuereinheit **16**, wenn die Prioritätsreihenfolge der ausgeführten Programme basierend auf Bedieneranweisungen, die von einem Steuerpult eingegeben werden, oder basierend auf Anweisungen von anderen Programmen dynamisch geändert wird.

[0083] Die Ausführungsprioritätssequenzsteuereinheit **16** ist immer darüber informiert welche Programme momentan ausgeführt werden und über die Prioritätsreihenfolge von diesen Programmen. Wenn es basierend auf Bedieneranweisungen, die von einem Steuerpult eingegeben werden, oder basierend auf Anweisungen von einem anderen Programm eine Anforderung gibt, die Prioritätsreihenfolge eines ausgeführten Programms zu ändern (S71), ist die Ausführungsprioritätssequenzsteuereinheit **16** in der Lage basierend auf der neuen Prioritätsreihenfolgeänderungsanforderung, die von einer externen Quelle empfangen wurde, die geeignete Prioritätsreihenfolge und die Beziehung zwischen den verschiedenen ausgeführten Programmen (S72) festzulegen.

[0084] Diese Auswertung vergleicht die Prioritätsreihenfolge des angeforderten Programms (genauer, das Programm für das die Änderung der Prioritätsreihenfolge angefordert wurde) mit der Prioritätsreihenfolge der momentan ausgeführten Programme, um die relative Priorität der betreffenden Programme festzulegen und um die Menge der Änderung, die in der Ausführungspriorität gemacht werden kann, einzustellen. Daher wird, wenn ein anderes Programm mit einer höheren Priorität ausgeführt wird und die Priorität des angeforderten Programms erhöht werden soll, die Priorität deutlich erhöht, um eine bevorzugte Ausführung gegenüber den anderen Programmen zu ermöglichen. Eine Änderung in der Prioritätsreihenfolge der Programme, die im Speicher ausgeführt werden, wird durch Ändern der Zeit, die die Programme die CPU (CPU) benutzen können, erreicht.

[0085] Somit wird die Prioritätsreihenfolge basierend auf dieser Verarbeitung (S73) geändert. Nachdem die Prioritätsreihenfolge geändert wurde, wird die neue Prioritätsreihenfolge sofort gültig und alle Programme werden basierend auf dieser neuen Prioritätsreihenfolge ausgeführt. Dieser Prozeß kann durch Mitteilen des Programmnamens und der geänderten (aktualisierten) Prioritätsreihenfolge an das Betriebssystem erreicht werden.

[0086] Die Ausführungsprioritätssequenzsteuereinheit **16** kann auch die Prioritätsreihenfolge der vorgemerkten Programme basierend auf Bedieneranwei-

sungen, die von einem Eingabepult eingegeben werden, oder basierend auf Anweisungen von einem anderen Programm dynamisch ändern. Wenn es eine Anforderung gibt, die Prioritätsreihenfolge der vorgemerkten Programme basierend auf Bedieneranweisungen, die von einem Eingabepult eingegeben werden, oder basierend auf Anweisungen von einem anderen Programm zu ändern, wird die Prioritätsreihenfolge des angeforderten Programmes mit der der anderen vorgemerkten Programmen verglichen, um, wie oben beschreiben eine Änderung der Prioritätsreihenfolge zu erwirken. Nachdem die Prioritätsreihenfolge geändert wurde, werden die vorgemerkten Programme in absteigender Ordnung der Priorität neu sortiert.

[0087] **Fig. 8** ist ein Flußdiagramm des Betriebes des Fehlerwiederherstellungsprozessors **18** in dem Computerbetriebsverwaltungssystem nach der vorliegenden Erfindung.

[0088] Wie in **Fig. 8** gezeigt ist, wird der Fehlerwiederherstellungsprozessor **18** aktiviert, wenn er durch das Ausführungsstatusüberwachungsmittel **14** informiert wird, daß ein Anwendungsprogramm in einem Fehlerzustand ist (S81). Der Fehlerwiederherstellungsprozessor **18** liest dann die Fehlerwiederherstellungsprozedur von der Programmdefinitionsdatei (S82).

[0089] Die Fehlerwiederherstellungsprozedur ist für jede Anwenderprogramm-kategorie getrennt definiert und enthält die Maßnahmen, die durchzuführen sind, wenn ein Fehler auftritt, die Wiederherstellungsprozedur, die Alarmbenachrichtigung, die Post, und die Maßnahmen, die für jedes verbundene Programm durchzuführen sind. Die gleiche Fehlerwiederherstellungsprozedur wird für alle Programme in einer einzelnen Programm-kategorie verwendet, obwohl es möglich ist eine einzelne Fehlerwiederherstellungsprozedur für jedes Programm zu definieren. Diese Daten können auf Anforderung des Systemverwalters geeignet neu geschrieben werden und die neuen Fehlerwiederherstellungsprozeduren können als die gültige Wiederherstellungsprozedur von dem Zeitpunkt an, ab dem die Prozedur neu geschrieben wurde, ausgeführt werden.

[0090] Der Fehlerwiederherstellungsprozessor **18** führt genau die Fehlerwiederherstellungsprozeduren (S84), wie definiert, aus. Diese Fehlerwiederherstellungsprozedur wird auch mit Vorrang gegenüber allen anderen ausgeführten Programmen ausgeführt und eine Aufnahme der Wiederherstellungsprozedurausführung wird als das Fehlerprotokoll, das den Computernamen, den Programmnamen, die ausgeführte Tätigkeit und die Zeit enthält in der Protokoll-datei **23** gespeichert (S83).

[0091] **Fig. 9** ist eine Tabelle eines Beispiels der Prozedurdefinitionen für die Fehlerwiederherstellungsverarbeitung in dem Computerbetriebsverwaltungssystem entsprechend der vorliegenden Erfindung. In **Fig. 9** beschreibt die Programm-kategorie den Programmnamen. Mehrere Kopien des gleichen

Programmes können gleichzeitig ausgeführt werden, aber die Fehlerwiederherstellungsprozedur ist die gleiche für alle ausgeführten Kopien des Programms. [0092] Die auszuführende Maßnahme, wenn ein Fehler auftritt (Spalte **2** der Tabelle), beschreibt welche Maßnahme aufzunehmen ist, wenn ein Fehler in einer Ausführung der Anwendungsprogramme auftritt. Die gezeigten Maßnahmen sind nichts Unternehmen, zwangsweises Beenden und eine Fehlerwiederherstellungsprozedur. Die Prozeduren für ein zwangsweises Beenden und die Fehlerwiederherstellungsprozeduren sind in der Programmdefinitionsdatei **21** definiert.

[0093] Die Alarmbenachrichtigung (Spalte **4**) definiert ob der Bediener automatisch mittels eines Funkrufs über einen Taschenempfänger (Pager), eines Funkrufs über das öffentliche Adressensystem oder anderen Mitteln informiert wird, wenn ein Fehler auftritt. Post (Spalte **5**) bezieht sich auf elektronische Post und definiert, ob ein Alarm (Fehlernachricht) auf dem Computerterminal angezeigt werden soll, wenn ein Fehler auftritt, oder ob eine e-mail Nachricht zu einem anderen Computerterminal des Netzwerkes gesendet werden soll. Die Kommunikationsfunktion, die angewendet wird, wenn ein Fehler auftritt, wird mittels der unten beschriebenen Fehlerkommunikationsfunktion erreicht.

[0094] Die verbundenen Programme (Spalte **6**) definieren die Programmgruppe, die gleichzeitig mit dem Programm ausgeführt wird, das den Fehler erzeugt, und die eine Beziehung mit diesem Programm aufweisen, oder sie definieren die Programmgruppe, die nach dem Beenden des Programms, das den Fehler erzeugt, ausgeführt werden soll. Die Fehlerwiederherstellungsprozedur ist in der Programmdefinitionsdatei **21** für alle Programme definiert, die zur der gleichen Gruppe gehören, und alle ausgeführten Programme in dieser Gruppe werden entsprechend der gleichen Fehlerwiederherstellungsprozedur verarbeitet.

[0095] **Fig. 10** ist ein Flußdiagramm des Betriebes des Endprozessors **17** in dem Computerbetriebsverwaltungssystem entsprechend der vorliegenden Erfindung.

[0096] Wenn eine Ausführung eines Programms gestoppt wird (S101), benachrichtigt das Betriebssystem den Endprozessor **17** des Prozeßverwalters, daß das Programm beendet wurde (S102). Basierend auf dem Endstatus des Programms, das vom Betriebssystem empfangen wurde, stellt der Endprozessor **17** fest, ob das Ende normal oder abnormal war (S103). Wenn das Programm aufgrund eines abnormalen Endes beendet wurde, wird der Fehlerwiederherstellungsprozessor **18** benachrichtigt (S107) und der entsprechende Wiederherstellungsprozeß wird angefordert. Wenn die Programmbeendigung normal war, speichert der Endprozessor **17** den Programmnamen des beendeten Programms, den Maschinennamen, die Zeit und das Datum, die Programmnummer und den Endstatus in das Endproto-

koll der Protokolldatei **23** (S104). Eine Prozedur, die zur Ausführung nach einer Programmbeendigung definiert ist, wird dann aus der Programmdefinitionsdatei **21** (S105) gelesen und die Prozedur wird genau ausgeführt (S106). Eine kontinuierliche Verarbeitung eines Programms kann unter Verwendung dieser Nachprozeßdefinition erreicht werden. Wenn der Nachprozeß durchgeführt ist, ist die komplette Reihenfolge der Prozesse durchgeführt.

[0097] **Fig. 11** ist ein Flußdiagramm des Betriebes des Computer-Ressourcen-Verwalters **15** in dem Computerbetriebsverwaltungssystem gemäß der vorliegenden Erfindung.

[0098] Wie in **Fig. 11** gezeigt ist, überprüft der Computer-Ressourcen-Verwalter **15** die aufsummierte bzw. gesamte CPU-Zeit, die durch jedes Programm verwendet wurde, für alle ausgeführten Programme, die durch den Prozeßverwalter **12** verwaltet werden (S111) und vergleicht dann die gesamte aktuelle CPU-Zeit mit den Steuergrenzen (vordefinierter Wert), die aus der Programmdefinitionsdatei (**5112**) ausgelesen wurden. Wenn der Vergleich zeigt, daß die Steuergrenze überschritten wird, werden der Fehlerwiederherstellungsprozessor **18** und das Fehlerkommunikationsmittel **19** benachrichtigt (S121, S122).

[0099] Wenn die aufsummierte CPU-Zeit die Steuergrenze nicht überschreitet, überprüft der Computer-Ressourcen-Verwalter **15** die Menge des verwendeten Speichers (S114) und vergleicht dann den aktuell verwendeten Speicher mit der Steuergrenze (vordefinierter Wert), die aus der Programmdefinitionsdatei gelesen wurde (S115). Wenn dieser Vergleich zeigt, daß die Steuergrenze überschritten wird, werden der Fehlerwiederherstellungsprozessor **18** und das Fehlerkommunikationsmittel **19** benachrichtigt (S121, S122).

[0100] Die Größe des verwendeten Diskettenplatzes wird in gleicher Weise überprüft (S117) und der aktuell verwendete Diskettenplatz wird mit den Steuergrenzen (vordefinierte Werte), die von der Programmdefinitionsdatei gelesen wurden (S118), verglichen. Wenn dieser Vergleich zeigt, daß die Steuergrenze überschritten wird, werden der Fehlerwiederherstellungsprozessor **18** und das Fehlerkommunikationsmittel **19** benachrichtigt (S121, S122).

[0101] Die Ressourcensteuergrenzen sind in der Programmdefinitionsdatei **21** durch den Benutzer definiert als der maximale Platz eines gegebenen Betriebsmittels, der durch jede Programmkategorie verwendet werden kann. In diesem Beispiel sind die aufsummierte CPU-Zeit, die Speicherzuweisung und die Diskettenplatzzuweisung speziell definiert. Die auszuführenden Aktionen, wenn die definierten Betriebsmittelgrenzen überschritten werden, werden auch durch den Benutzer für jedes Programm (oder Programmkategorie) in der Programmdefinitionsdatei **21** definiert.

[0102] Der Netzwerkverwalter **20** des Computerbetriebsverwaltungssystems gemäß der vorliegenden

Erfindung wird im folgenden beschrieben.

[0103] Die Aufgabe des Netzwerkverwalters **20** ist es den Betrieb eines großen Anwendungsprogramms zu verwalten, das auf einer Mehrzahl von Computern durch Kommunikation zwischen den einzelnen Kopien der Anwendung (eine Netzwerkanwendung) betrieben wird, dadurch daß die Prozeßverwalter **12** in die Lage versetzt werden miteinander zu kommunizieren und Informationen zwischen mehreren Netzwerkcomputern, die miteinander zu einem gemeinsamen Netzwerk verbunden sind, zu teilen und verwendet werden, um individuell eine Anwendung auszuführen, die mit anderen Kopien dieser Anwendung, die auf anderen Computern laufen, kommuniziert.

[0104] Die Information, die zwischen den Prozeßverwaltern **12** auf verschiedenen Computern ausgetauscht werden, ist die Protokolldateiinformatio, die sich auf den Programmstart, -ende und -fehler bezieht; sind die Betriebsstatusdaten für Anwendungsprogramme, die durch Kommunikation zwischen den Computern arbeiten; und sind Informationen, die die Programmdefinitionsdatei (einschließlich Prioritätsreihenfolge und Fehlerwiederherstellungsprozeduren) betreffen. Die Protokolldatei speichert auch den Namen des Computers von dem die Informationen stammen, wodurch es möglich ist, festzustellen von welchem Computer die Protokolldatei stammt und wodurch es ermöglicht wird, daß die Informationen durch einen einzelnen Computer überwacht werden. Der Betriebsstatus der Anwenderprogramme, die während der Kommunikation mit Programmen, die auf anderen Netzwerkcomputern ausgeführt werden, arbeiten, wird durch die Prozeßverwalter, die auf den einzelnen Computern ausgeführt werden, verwaltet. Die einzelnen Prozeßverwalter kommunizieren dann miteinander regelmäßig, so daß die Netzwerkanwendung ohne Konflikt ausgeführt werden kann. Die Programmprioritätsreihenfolge, die Fehlerwiederherstellungsprozedur und andere Informationen in der Programmdefinitionsdatei werden zentral verwaltet, dadurch daß die Informationen, die in einen Computer eingegeben werden, zu mehreren Computern – wie notwendig – kopiert werden.

[0105] **Fig. 12** ist ein Flußdiagramm des Betriebes des Netzwerkverwalters **20** in dem Computerbetriebsverwaltungssystem gemäß der vorliegenden Erfindung.

[0106] Wie in **Fig. 12** gezeigt ist, versendet der Prozeßverwalter **12** eine Statusanforderungsnachricht an das Anwendungsprogramm, das auf einem Computer ausgeführt wird, um den Status dieses Programms zu überwachen (S121). Wenn der Prozeßverwalter **12** die Antwort von diesem Anwendungsprogramm innerhalb der vordefinierten Zeitdauer empfängt (S122), setzt der Prozeßverwalter **12** die Überwachung des Programmstatus durch Wiederholen dieser Nachrichtenschleife, wie oben beschrieben, fort.

[0107] Wenn der Prozeßverwalter **12** die Antwort

des Anwendungsprogramms nicht innerhalb der vordefinierten Zeitdauer empfängt (S122), überprüft der Prozeßverwalter **12** den Ausführungsstatus des abgefragten Programms. Der Prozeßverwalter **12** erkennt den Kommunikationscomputer und das Programm basierend auf der Definition der verbundenen Programme in der Programmdefinitionsdatei **21**. Nachrichten werden zwischen den Prozeßverwaltern von den kommunizierenden Computern zum Statusüberwachen ausgetauscht (S123). Wenn der Prozeßverwalter die Antwort von dem Programm normal empfängt (S124), stellt der Prozeßverwalter fest, daß ein Fehler auf dem Netzwerk aufgetreten ist (S127). Wenn der Prozeßverwalter die Antwort von dem Programm nicht empfängt, stellt der Prozeßverwalter fest, daß ein Fehler in einem der verwalteten Programme aufgetreten ist (S125) und schreibt ein Fehlerprotokoll in die Protokolldatei **23** (S126).

[0108] **Fig. 13** und **14** sind Flußdiagramme des Betriebes des Fehlerkommunikationsmittels **19**. Das Fehlerkommunikationsmittel **19** hat zwei Funktionen: (1) eine Fehlerkommunikationsfunktion, die den Bediener gemäß der Prozedur, die zum Ausführen definiert ist, wenn ein Fehler auftritt, benachrichtigt, daß ein Fehler aufgetreten ist, (2) eine Fehlerüberwachungsfunktion zum Feststellen des Auftretens von Fehlern auf einem anderen Computer des Netzwerkes.

[0109] Das Flußdiagramm in **Fig. 13** beschreibt die Fehlerkommunikationsfunktion. Wenn das Fehlerkommunikationsmittel **19** durch das Statusüberwachungsmittel **14** oder den Computer-Ressourcen-Verwalter **15** von einem Fehler benachrichtigt wird (S131), wird die Fehlerwiederherstellungsprozedur von der Programmdefinitionsdatei **21** gelesen (S132). Um den Bediener von dem Fehler zu benachrichtigen, werden die Nachrichteninhalte zu einem allgemeinen Zweckalarm, einer elektronischen Post, einer zentralen Überwachungsstation oder anderen Mitteln, die durch die Fehlerwiederherstellungsprozedur definiert sind, ausgegeben (S133).

[0110] Eine allgemeiner Zweckalarm ist im wesentlichen ein Zwischencomputer- oder öffentliches Adressensystem, wobei eine Anzeige unter Verwendung von elektronischen Tönen gemacht wird. Elektronische Post bezieht sich auf elektronische Postnachrichten, die anzeigen das ein Fehler aufgetreten ist. Diese Nachrichten werden an dem Computerterminal des Bedieners, des Systemverwalters oder anderen definierten Bestimmungsorten ausgegeben. Die zentrale Überwachungsstation ist einfach eine zentrale Computersystemeingabeeinheit oder Anzeige, wo Nachrichten angezeigt werden. Es ist auch möglich gleichzeitig die gleiche Nachricht in eine Diskettendatei oder einen Drucker auszugeben.

[0111] Das Flußdiagramm in **Fig. 14** beschreibt den Betrieb, der durch die Fehlerkommunikationsfunktion ausgeführt wird nach der Fehlerfeststellung durch die Fehlerüberwachungsfunktion des Fehlerkommunikationsmittels **19**. Die Fehlerüberwachungsfunktion

überwacht regelmäßig den Inhalt der Protokolldateien **23** der Netzwerkcomputer, um zu überprüfen, ob ein neues Fehlerprotokoll erzeugt worden ist (S141). Wenn ein neues Fehlerprotokoll nicht festgestellt wird (S142), führt die Fehlerüberwachungsfunktion die regelmäßige Überprüfung für neue Fehlerprotokolle fort (S141). Wenn ein neues Fehlerprotokoll festgestellt wird (S142), wird der Protokollinhalt überprüft (S143). Wenn ein Fehlerprotokoll durch die Fehlerüberwachungsfunktion bestätigt wird, wird die Fehlerwiederherstellungsprozedur von der Programmdefinitionsdatei **21** durch die Fehlerkommunikationsfunktion gelesen (S144) und ein Alarm wird entsprechend der in der Programmdefinitionsdatei **21** vordefinierten Prozedur ausgegeben (S145).

[0112] Somit verwaltet das Fehlerkommunikationsmittel **19** die Protokolldateien der Netzwerkcomputer über die Prozeßverwalter **12** der Netzwerkcomputer zentral und es umfaßt eine Funktion zur Kommunikation der Fehler und eine Funktion zum zentralen Überwachen, ob Fehler auf dem Netzwerk aufgetreten sind. Die Fehlerüberwachungsfunktion wird durch Feststellen eines neuen Fehlerprotokolls auf einem Netzwerkcomputer ausgelöst. Die Fehlerüberwachungsfunktion überwacht definierte Typen von Fehlerprotokollen, die auf einem Computer erzeugt wurden, innerhalb eines spezifischen Bereiches, der in der Programmdefinitionsdatei vorher festgelegt ist.

2. Ausführungsform

[0113] **Fig. 15** ist ein Flußdiagramm und ein Blockdiagramm von den verschiedenen Steuereinheiten in einer zweiten Ausführungsform eines Computerverwaltungsbetriebssystems gemäß der vorliegenden Erfindung.

[0114] Wie in **Fig. 3** gezeigt ist, umfaßt die erste Ausführungsform der vorliegenden Erfindung die Computerressourcenverwaltungsfunktion als Teil der Hauptprogrammenschleife. In der in **Fig. 15** gezeigten zweiten Ausführungsform arbeitet jedoch der Computerressourcenverwalter als ein unabhängiges Programm, das auf Anfrage aufgerufen und ausgeführt wird, wie durch das Hauptprogramm verlangt wird.

[0115] Die verschiedenen in **Fig. 15** gezeigten Steuereinheiten arbeiten identisch zu denen, die in der obigen ersten Ausführungsform beschrieben wurden.

[0116] Wie in **Fig. 15** gezeigt ist, geht die Ausführungsanforderung zu dem Computerbetriebsverwaltungssystem **1** und nicht zu dem Betriebssystem **2** in einem Computerbetriebsverwaltungssystem **1** gemäß der vorliegenden Erfindung, wenn ein Anwendungsprogramm **3** ausgeführt werden soll. Wenn eine Programmstartanforderung durch den Bediener oder von einem anderen Programm zu den Computerbetriebsverwaltungssystem **1** ausgegeben wird (S201), liest die Ausführungssteuereinheit **13** die maximale Anzahl der gleichzeitig ausführbaren Programme von der Programmdefinitionsdatei **201**

(S202) und vergleicht diese maximale Anzahl mit der Anzahl der ausgeführten Programme. Wenn die Anzahl der ausgeführten Programme kleiner ist als die gelesene Grenze (maximale Anzahl), wird eine Tochterprogrammstartanforderung zu dem Betriebssystem ausgegeben (S203). Wenn die Anzahl der ausgeführten Programme gleich zu der Grenze ist (maximale Anzahl) und eine Programmstartanforderung ausgegeben wird, wird das angeforderte Programm in der Aufnahmedatei für vorgemerkte Programme **203** vorgemerkt. Die Ausführungssteuereinheit **13** überwacht regelmäßig die Anzahl der ausgeführten Programme. Als ein Ergebnis wird, wenn es möglich wird, ein vorgemerktes Programm auszuführen, das nächste auszuführende Programm von der Aufnahmedatei für vorgemerkte Programme **203** gelesen und eine Startanforderung wird zu dem Betriebssystem ausgegeben (S203). Die Ausführungssteuereinheit **13** speichert dann das Startprotokoll in die Protokolldatei **202**, wenn der Programmstart beendet ist.

[0117] Wenn ein Programm einmal gestartet ist, überwacht das Ausführungsstatusüberwachungsmittel **14** den Ausführungsstatus von jedem ausgeführten Programm durch Senden von Nachrichten zu und Empfangen von Antworten von jedem ausgeführten Programm (S204). Wenn ein Fehler bei einem überwachten Programm auftritt, werden der Fehlerwiederherstellungsprozessor **18** und das Fehlerkommunikationsmittel **19** genau benachrichtigt, um die vordefinierten Prozesse auszuführen (S209 und S210).

[0118] Wenn es von dem Bediener oder einem anderen Programm während der Programmausführung eine Anforderung gibt, die Prioritätsreihenfolge eines ausgeführten Programmes oder eines vorgemerkten Programmes in der Aufnahmedatei für vorgemerkte Programme (Anforderungswarteschlange) **203** zu ändern, ändert die Ausführungsprioritätssequenzsteuereinheit **16** die Prioritätsreihenfolge der ausgeführten Programme oder der Programme in der Aufnahmedatei für vorgemerkte Programme **203** (S207).

[0119] Soweit notwendig benachrichtigt das Ausführungsstatusüberwachungsmittel **14** den Computer-Ressourcen-Verwalter **15** und der Computer-Ressourcen-Verwalter **15** überwacht die Computerressourcen bzw. Betriebsmittel (S208) durch Vergleichen der aktuellen Verwendung der Computerbetriebsmittel (einschließlich der CPU-Zeit, Speicher und des Diskettenlaufwerkes) mit den Betriebsmittelverwendungsgrenzen, die in der Programmdefinitionsdatei **21** gespeichert sind. Wenn ein Fehler auftritt, werden der Fehlerwiederherstellungsprozessor **18** und das Fehlerkommunikationsmittel **19** benachrichtigt, um die vordefinierten Prozesse auszuführen (S209 und 210).

[0120] Die Ausführungsstatusüberwachung (S204) wird dann in einer kontinuierlichen Schleife ausgeführt, bis das Betriebssystem **2** den Prozeßverwalter **12** benachrichtigt, daß ein Programm beendet wurde. Der Netzwerkverwalter **20** überwacht auch Programme, die ausgeführt werden während Kommunikation

mit anderen Programmen auf dem Netzwerk mittels Austauschen von Informationen mit den Prozeßverwaltern **12** der anderen Computer **204** des Netzwerkes (S211).

[0121] Wenn ein Fehler während der Ausführungsstatusüberwachung oder der Computer-Ressourcen-Verwaltung auftritt, wird die Fehlerinformation in dem Fehlerwiederherstellungsprozessor **18** zur Verfügung gestellt, der den Wiederherstellungsprozess gemäß der in der Programmdefinitionsdatei gespeicherten Wiederherstellungsprozedur durchführt und der gleichzeitig das Fehlerprotokoll in die Protokolldatei **202** schreibt (S209). Das Fehlerkommunikationsmittel **19** gibt auch einen geeigneten Alarm oder eine andere Benachrichtigung gemäß des Fehlerwiederherstellungsprozesses aus, der in der Programmdefinitionsdatei gespeichert ist (S210).

[0122] Wenn der Prozeßverwalter **12** durch das Betriebssystem benachrichtigt wird, daß ein Programm beendet wurde, führt der Endprozessor **17** den Endprozeß aus, der in der Programmdefinitionsdatei **21** gespeichert ist, und schreibt das Endprotokoll in die Protokolldatei **202** (S206).

Vorteile der Erfindung

[0123] Das Computerbetriebsverwaltungssystem der vorliegenden Erfindung erreicht die folgenden Vorteile durch effektives Verwenden des Hauptspeichers (Speicher), des ergänzenden Speichers (Diskettenlaufwerk) und der CPU eines einzelnen Computers oder Netzwerkcomputern durch Begrenzen und Überwachung der Anzahl der gleichzeitig ausführbaren Programme für das Betriebssystem.

(1) Durch Begrenzen der Anzahl der gleichzeitig ausführbaren Programme für das Betriebssystem und durch Verwalten der Programmausführung, so daß Speichermangel oder Diskettenplatzmangel nicht auftritt, kann eine übermäßige Verschlechterung der Verarbeitungsgeschwindigkeit aufgrund Speichermangels verhindert werden und die Leistung kann daher verbessert werden, wenn mehrere Programme gleichzeitig auf einem einzelnen Computer oder Netzwerkcomputern ausgeführt werden.

(2) Gezwungenes abnormales Beenden von einigen Programmen und der Verlust von einiger Systemfunktionalität aufgrund des zeitweisen nicht ausreichenden Speichers oder Diskettenplatzes kann verhindert werden, wenn mehrere Programme gleichzeitig ausgeführt werden und ein sehr zuverlässiges System, das frei von normalen Programmbeendigungen ist, kann erreicht werden.

(3) Wenn in einem Anwendungsprogramm ein Fehler auftritt und das Programm abnormal endet, können das Programm, das die Schwierigkeit verursacht, und der Zeitablauf des Fehlers durch Bezug auf ein zentral verwaltetes Protokoll herausgefunden werden. Fehler die in Systemen auftreten, bei denen Programme über ein Netzwerk ver-

wendet werden oder bei denen mehrere Programme zusammen arbeiten, können auch herausgefunden werden.

(4) Es ist auch möglich Fehler festzustellen, die bei unvollständigen Programm auftreten, die keine Fehlermeldung ausgeben sogar wenn ein Fehler während der Programmausführung aufgetreten ist.

(5) Die Programmausführungspriorität kann definiert werden und damit kann eine Echtzeitverarbeitung ermöglicht werden, wenn sie durch ein gegebenes Programm verlangt wird. Es ist auch möglich die Programme gemäß ihrer Wichtigkeit zu gewichten, so daß wichtige Programme mit Priorität verarbeitet werden.

(6) Die Wiederherstellungsprozedur, die hier ausgeführt wird, wenn ein Programm abnormal endet, kann definiert werden und automatisch ausgeführt werden, und damit wird die Fehlersuche für den Bediener stark vereinfacht.

Patentansprüche

1. Computer

- mit einer zentralen Verarbeitungseinheit, einem Hauptspeicher und einem Zusatzspeicher;
- mit UNIX als Betriebssystem (2), wobei das Betriebssystem (2) den Computer in die Lage versetzt, mehrere Anwendungsprogramme (3) gleichzeitig auszuführen; und
- mit einem Computerbetriebsverwaltungssystem (1, 11), bestehend aus
 - einem Prozeßverwalter (12), der zwischen dem UNIX-Betriebssystem (2) und den mehreren Anwendungsprogrammen (3) liegt und der
 - einen Ausführungszustandsüberwacher (14) zum Überwachen von Ausführungszuständen der mehreren Anwendungsprogramme (3) dadurch, daß er periodisch eine Nachricht zu jedem Anwendungsprogramm (3) sendet und von den Inhalten der von jedem Anwendungsprogramm (3) zurückgeschickten Nachricht bestimmt, ob das Anwendungsprogramm (3) in Ausführung ist,
 - einen Computerbetriebsmittelverwalter (15) zum Überwachen eines Benutzerstatus der Computerressourcen,
 - einen Fehlerwiederherstellungsprozessor (18) zum Ausführen eines Fehlerwiederherstellungsprozesses, wenn ein Fehler durch den Computerbetriebsmittelverwalter (15) festgestellt worden ist, und
 - einen Endprozessor (17) zum Ausführen eines Nachprozesses, wenn der Ausführungszustandsüberwacher (14) erfaßt, daß eines der mehreren ausgeführten Anwendungsprogramme (3) beendet ist, aufweist; und
 - mit einem Verwaltungsprogramm, das als ein Anwendungsprogramm (3) vor dem Betriebssystem (2) agiert; sowie
 - mit einer Programmdefinitionsdatei (21) zum Speichern von Informationen betreffend der maximalen

Anzahl von gleichzeitig ausführbaren Anwenderprogrammen (3) in jeweils einer Programmkategorie, – bei dem das Verwaltungsprogramm die Anzahl der momentan ausgeführten Anwendungsprogramme (3) beim Starten eines neuen Anwendungsprogrammes (3) überwacht und das Starten des neuen Anwendungsprogrammes (3) verhindert, wenn die Anzahl der momentan ausgeführten Anwendungsprogramme (3) gleich der in der Programmdefinitionsdatei (21) gespeicherten maximalen Anzahl ist.

2. Computer nach Anspruch 1, bei dem der Ausführungszustand der Anwendungsprogramme durch Ausführen des Verwaltungsprogrammes in einer Art, die eine direkte Kommunikation mit den zu verwaltenden Anwendungsprogrammen (3) oder eine direkte Verwaltung der zu verwaltenden Anwendungsprogramme (3) ermöglicht, verwaltet und gesteuert wird.

3. Computer nach Anspruch 1 oder 2 weiter mit einer Ausführungsprioritätssteuereinheit (16) zum Steuern der Ausführungspriorität von jedem Anwendungsprogramm (3) für jede Programmkategorie.

4. Computer nach Anspruch 3, bei dem die Ausführungsprioritätssteuereinheit (16) die Prioritätsreihenfolge der Programme in einer Aufnahmedatei (22) für vorgemerkte Programme zum vorübergehenden Speichern von Daten der Anwendungsprogramme (3), die zum Ausführen vorgemerkt sind, steuert und eine Prioritätsreihenfolge der ausgeführten Programme basierend auf der Ausführungspriorität von jedem Anwendungsprogramm (3) bestimmt.

5. Computer nach Anspruch 1, bei dem der Fehlerwiederherstellungsprozessor (18) einen vorbestimmten Fehlerwiederherstellungsprozeß für ein Anwendungsprogramm (3) ausführt, wenn keine Nachricht von dem Anwendungsprogramm (3) zurückgesendet ist.

6. Computer nach Anspruch 1, bei dem der Endprozessor (17) einen Nachprozeß durchführt, wenn eine für ein Ende eines Anwendungsprogrammes (3) und einen Zustand des Endes bezeichnende Nachricht von dem Anwendungsprogramm (3) ausgegeben ist, wobei der Nachprozeß entsprechend dem Zustand des Endes vordefiniert ist.

7. Computer nach einem der Ansprüche 1 bis 6 weiter mit einem Fehlerfeststellungsmittel zum Feststellen eines Fehlers, wenn ein Anwendungsprogramm (3) ausgeführt wird, das eine der mit Bezug zu dem Haupt- und Zusatzspeicher und der aufsummierten Verarbeitungszeit, die dem Anwendungsprogramm (3) zugeordnet sind, vordefinierten Grenzen übersteigt.

8. Computer nach einem der Ansprüche 1 bis 8 weiter mit einem Fehlerzeichengeber (19) zum Mittei-

len eines Fehlers an einen Anwender, wenn der Fehler erfaßt ist.

9. Computersystem mit:
einer Mehrzahl von Computern **(204)** nach Anspruch 1, die durch ein Netzwerk verbunden sind, und einem Netzwerkverwalter **(20)** zum Kommunizieren zwischen den Prozeßverwaltern **(12)** von zwei beliebigen Computern **(204)**;
wobei eine Programmdefinitionsdatei durch einen Benutzer definierte Informationen für Programme, die gemeinsam über das Netzwerk ausgeführt sind, aufzeichnet, und wobei der Netzwerkverwalter **(20)** den Betrieb der gemeinsam ausgeführten Programme **(3)** entsprechend der in der Programmdefinitionsdatei **(21)** aufgezeichneten Informationen verwaltet.

Es folgen 14 Blatt Zeichnungen

Fig. 1

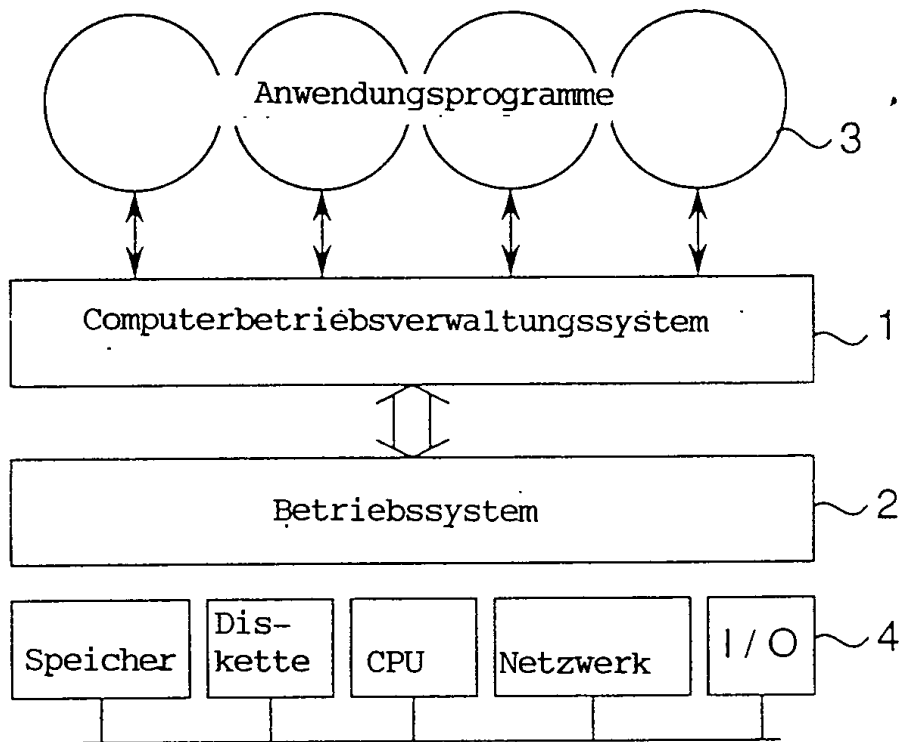


Fig.2

11 Computerbetriebsverwaltungssystem

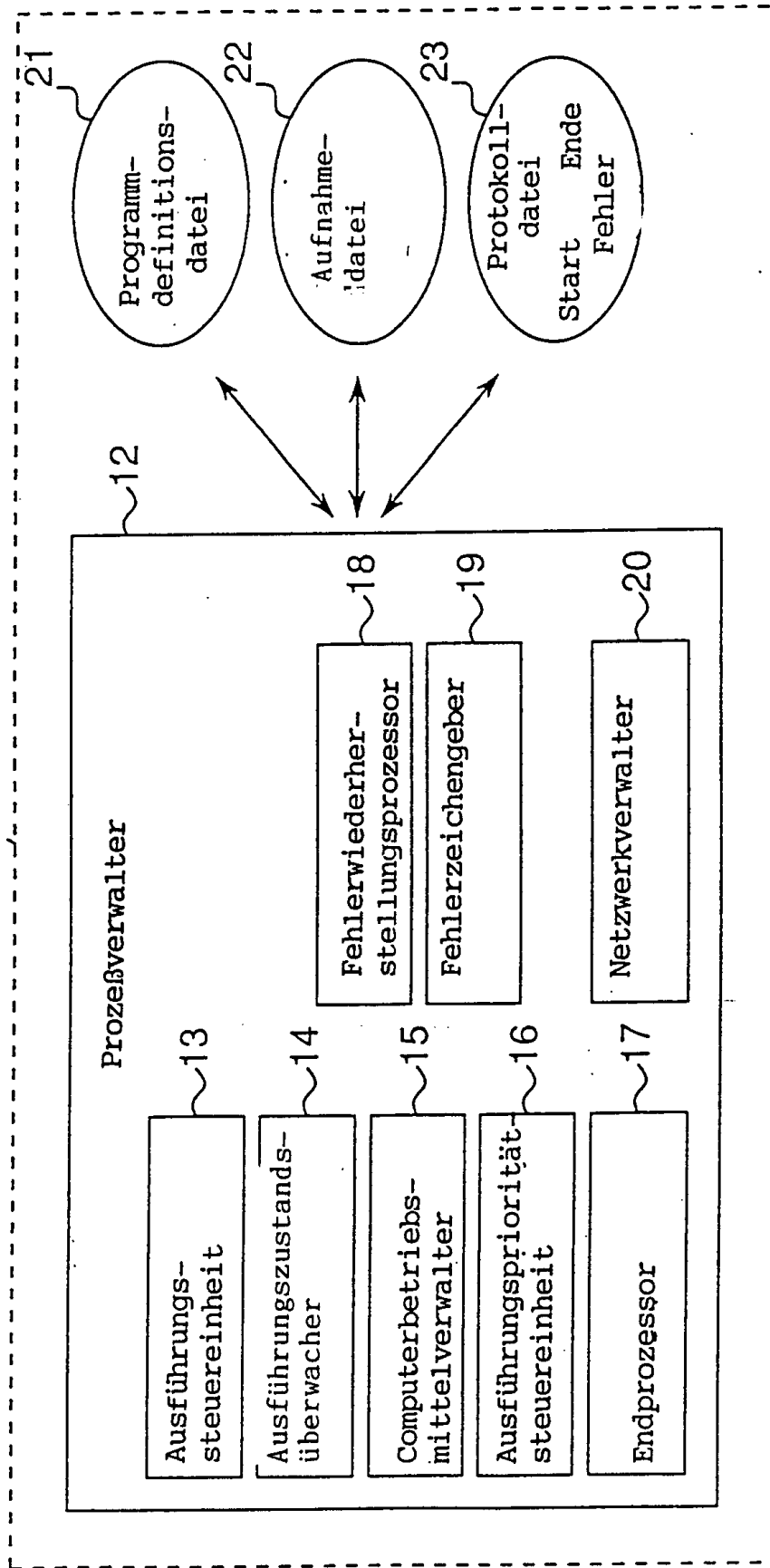


Fig.3

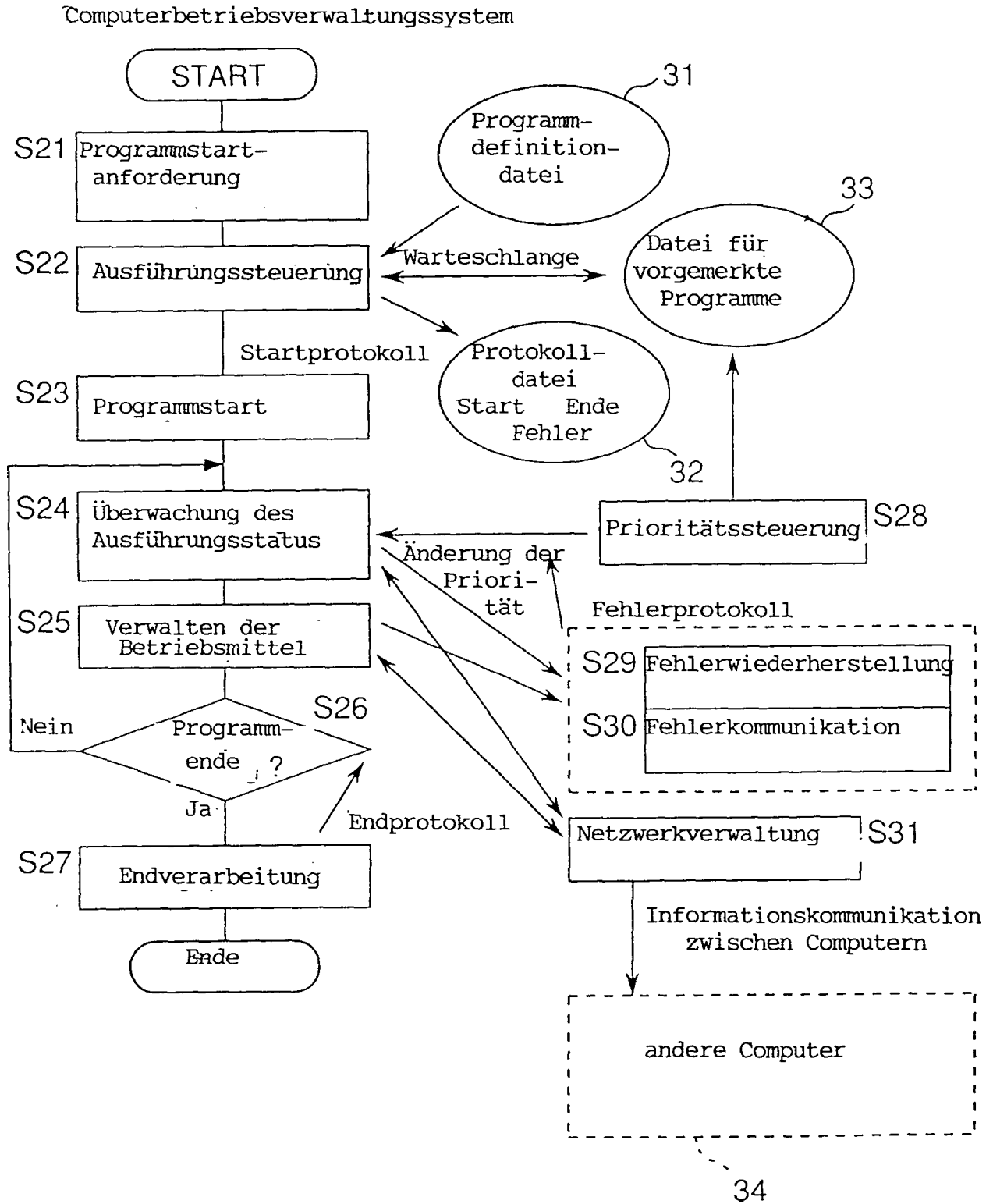


Fig.4

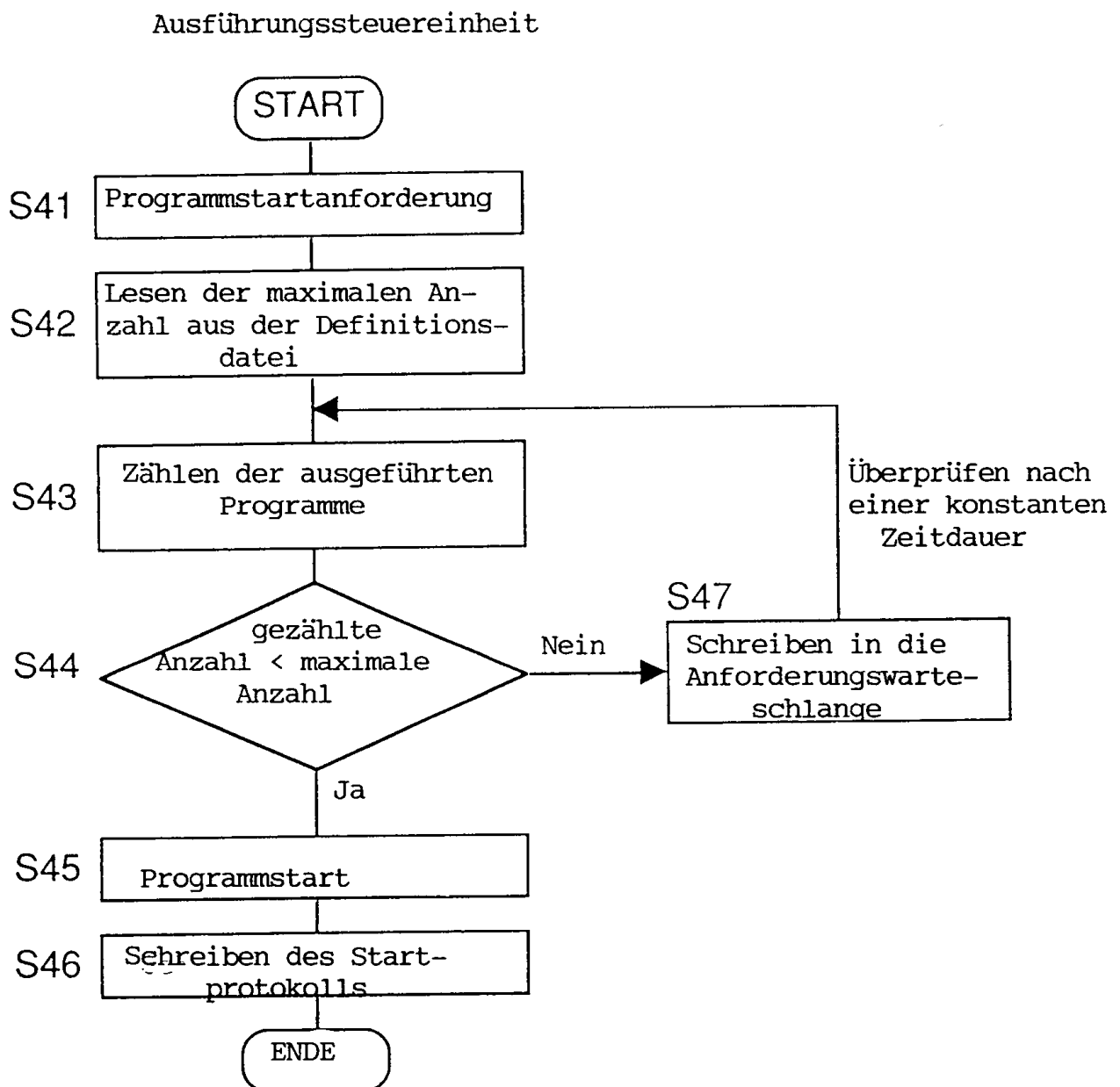


Fig.5

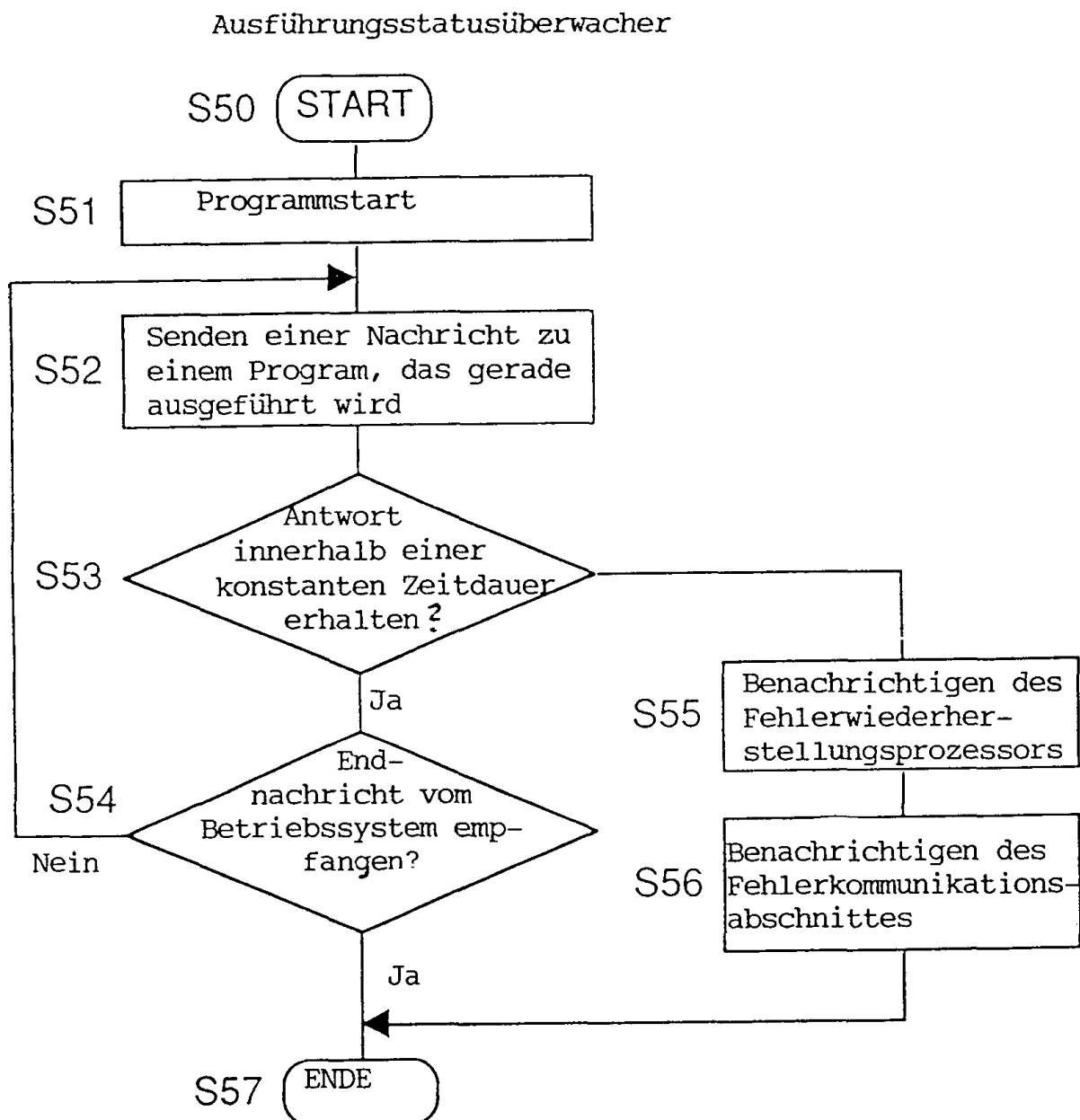


Fig.6

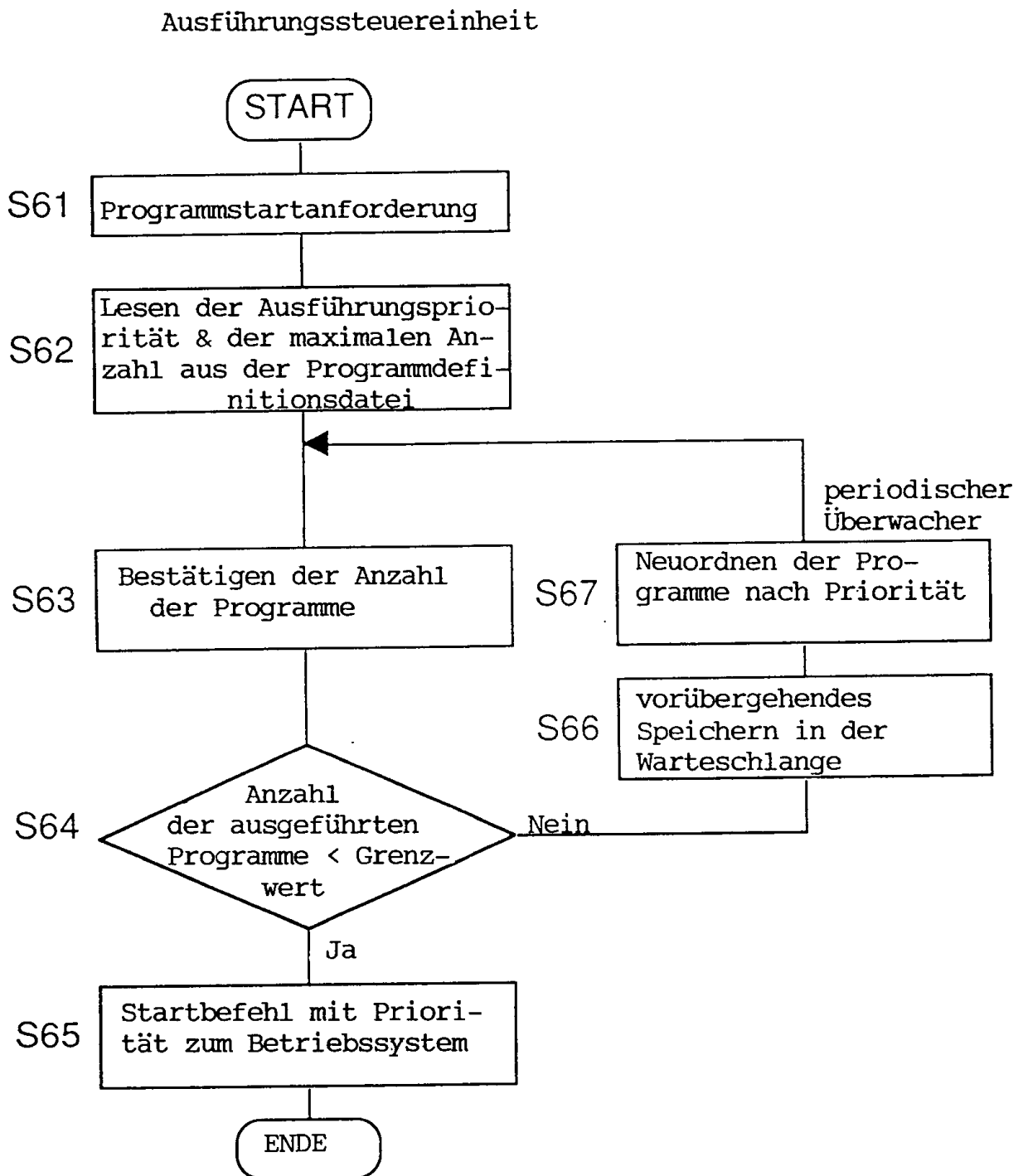


Fig. 7

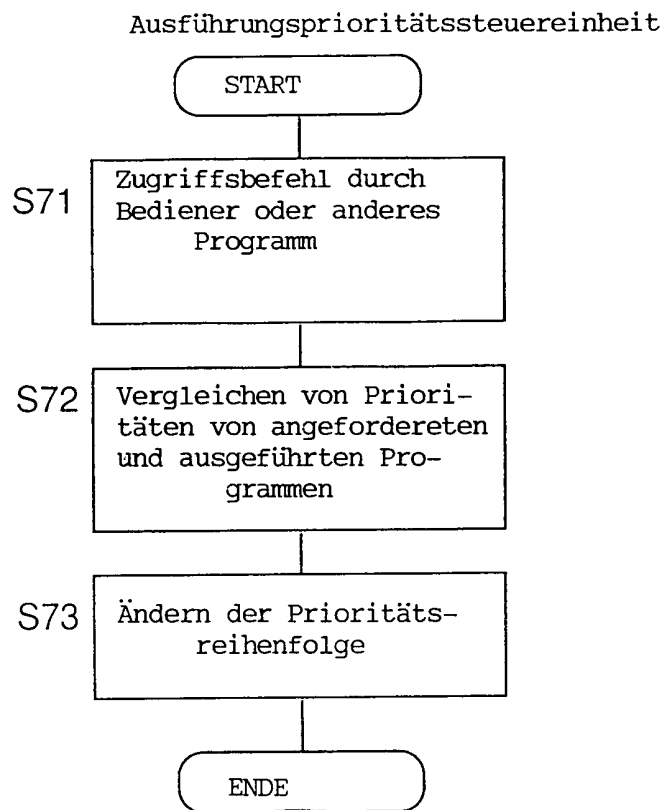


Fig. 8

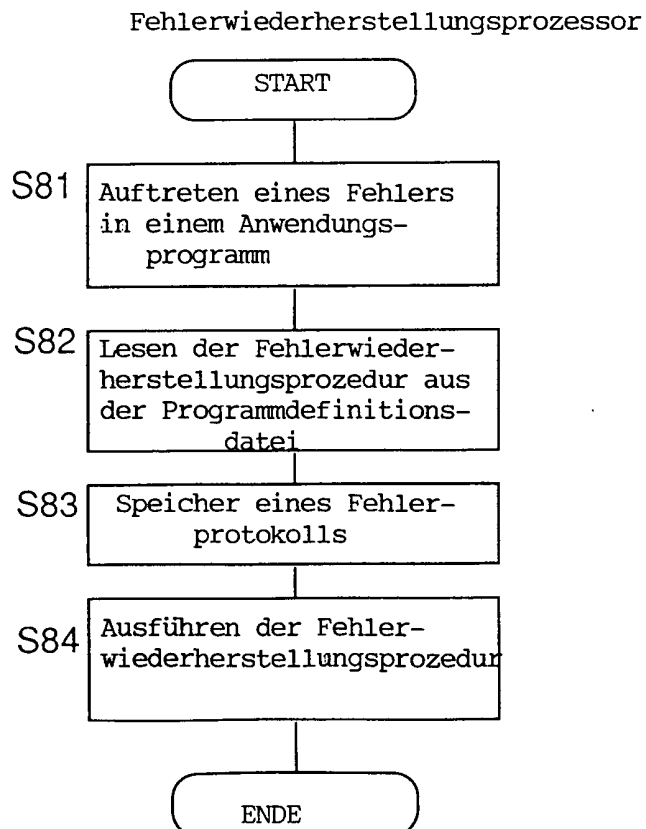


Fig.9

Definition der Fehlerwiederherstellungsprozedur

Programm- kategorie	Maßnahme bei einem Fehler	Prozedur	Alarm	Post	Verbundene Programme	Verarbeitung der verbundenen Programme
Anwendung 1	nein	nein	ja	nein	Gruppe 1	nein
Anwendung 2	Wiederherstel- lungsprozess	Prozedur 1	ja	Post	nein	Verarbeitung 4
Anwendung 3	zwangswises Beenden	Prozedur 7	nein	Post	Gruppe 2	Verarbeitung 7

Fig. 10

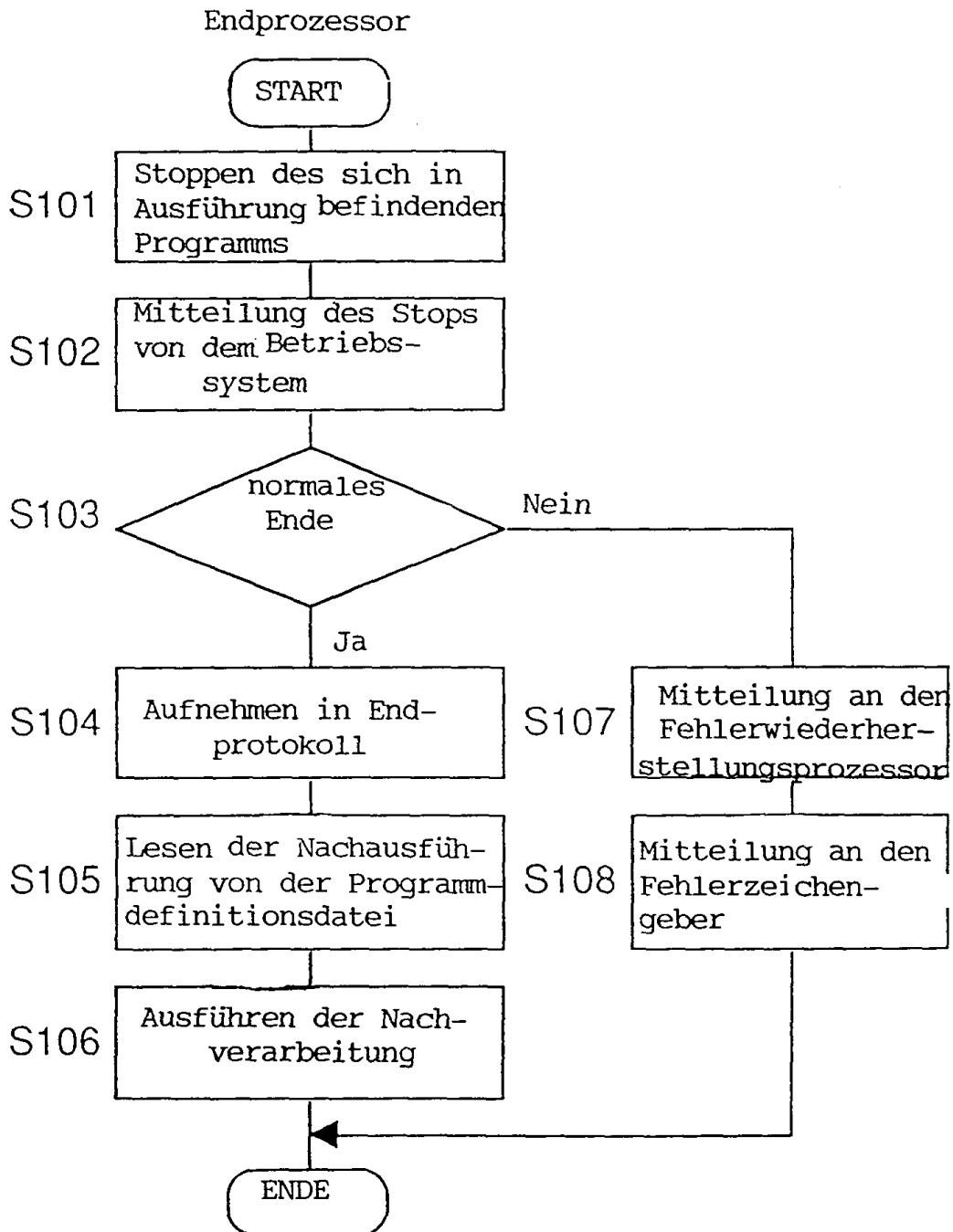


Fig. 11

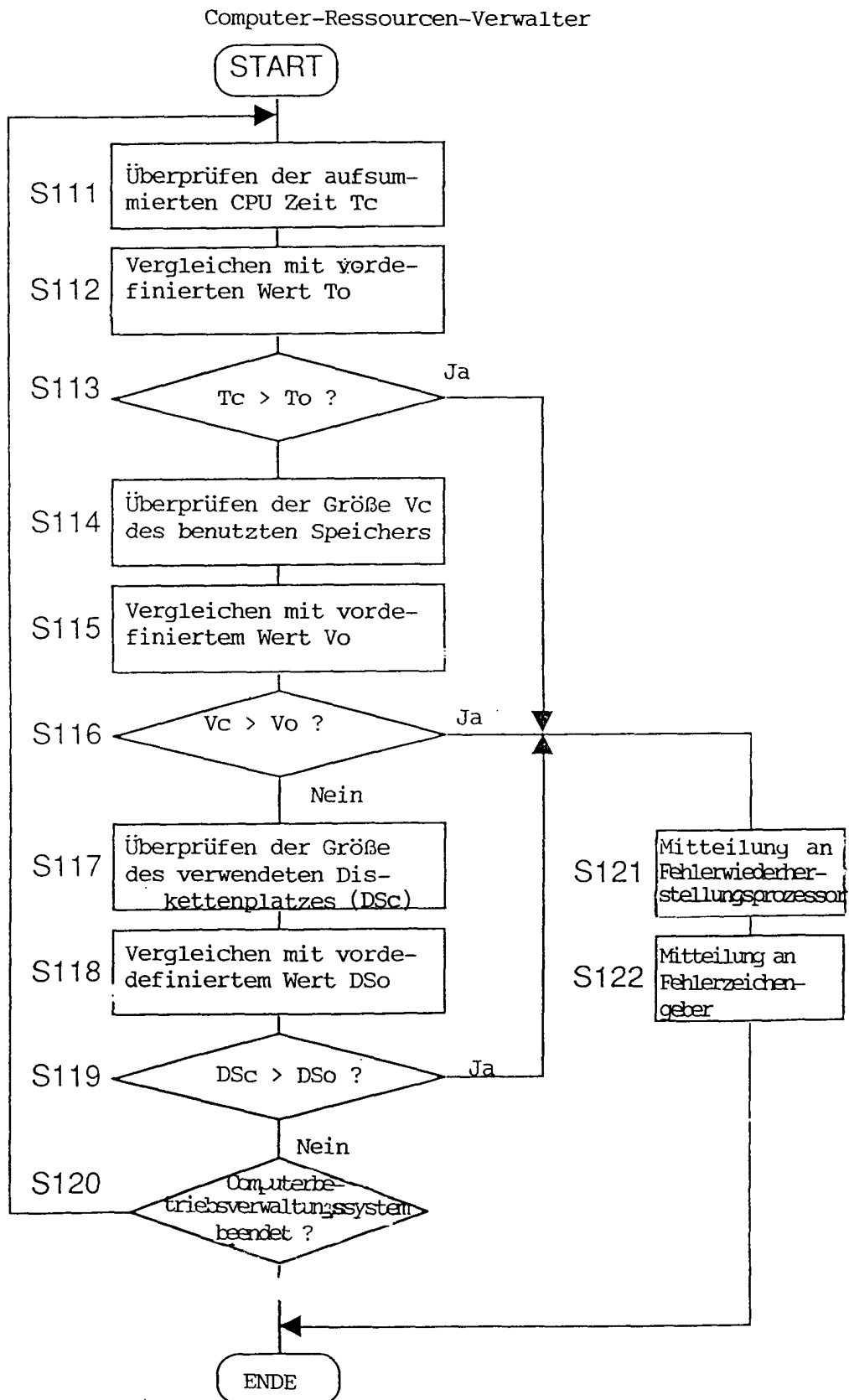


Fig.12

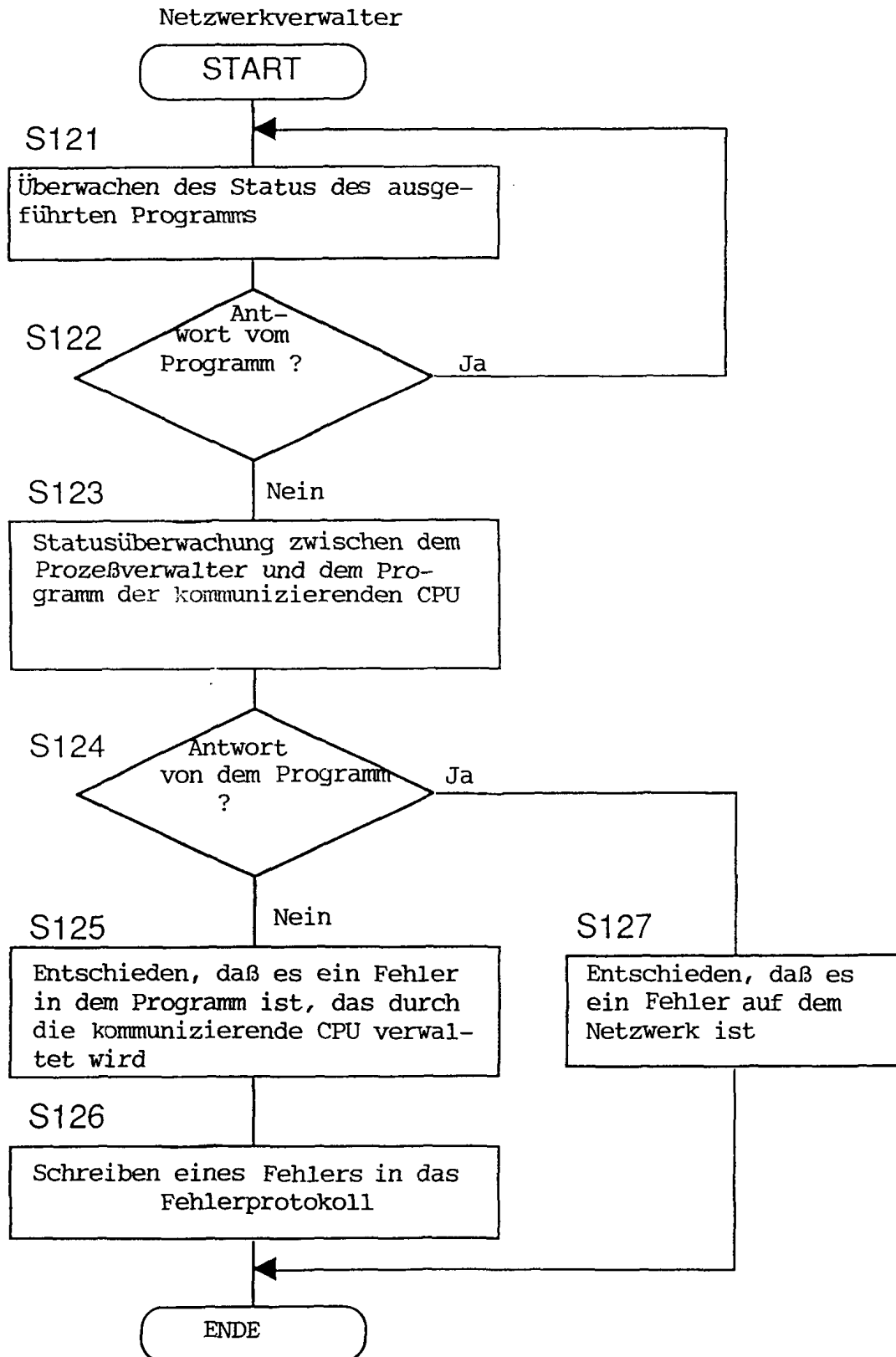


Fig. 13

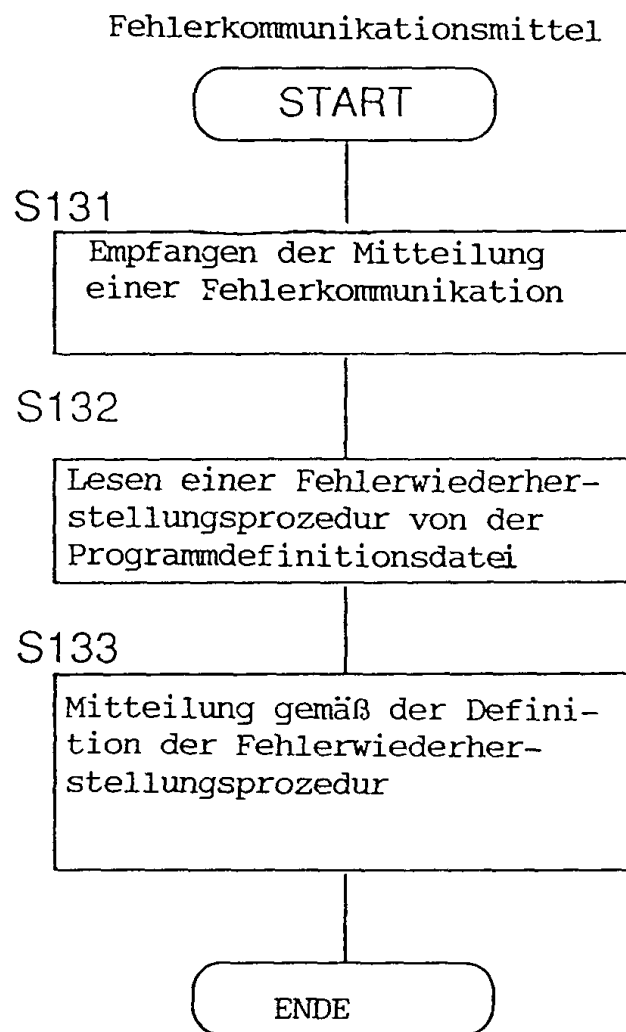


Fig. 14

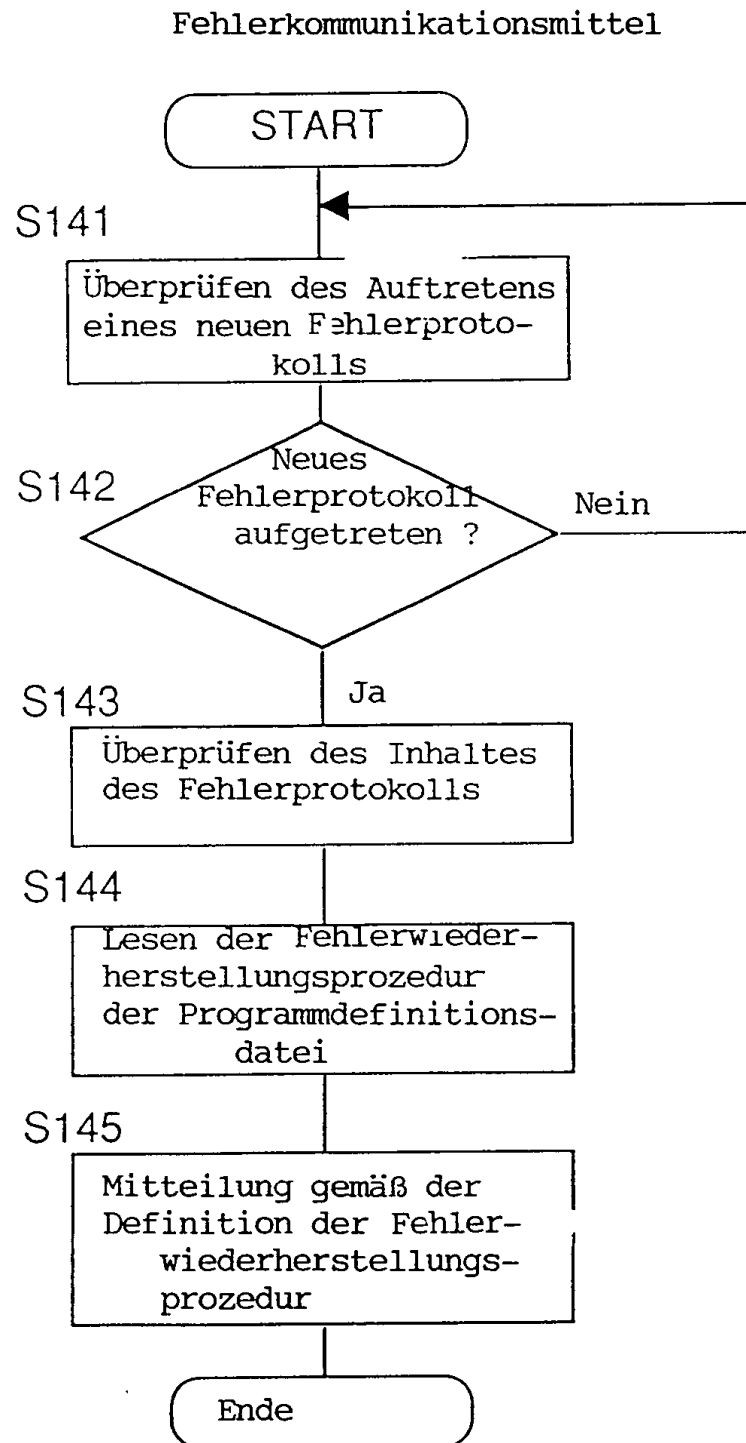


Fig. 15

