

## Anwenderaspekte des Systems virtueller Maschinen - SVM, Ausgabe 3.3

*Gottfried Gerbing; Dr. Klaus Heinecke VEB Robotron-Elektronik Dresden FG Geräte Karl-Marx-Stadt Bernd Lilpopp VEB Leitzentrum für Anwendungsforschung Berlin*

*Das Betriebssystem SVM 3.3 ist Bestandteil des Betriebssystemkomplexes OC-7.1/EC, der im Rahmen des ESER in internationaler Zusammenarbeit entwickelt wird. Dieser Komplex ist - vom Komplex OC-7 EC ausgehend/5/- die nächste Entwicklungsstufe der Betriebssysteme, die zur Auslieferung an die Anwender kommt. Zu diesem Betriebssystemkomplex gehören die Betriebssysteme SVS 7.1/6/, BPS 7.1 und SVM 3.3. Das Betriebssystem SVM ist dabei das Betriebssystem, das das Konzept der virtuellen Maschinen realisiert und einen leistungsfähigen Dialogbetrieb auf den virtuellen Maschinen sichert. Die Betriebssysteme BPS und SVS sind im wesentlichen für die Stapelarbeit vorgesehen. Das Betriebssystem BPS arbeitet **nur** auf virtuellen Maschinen des SVM, das heißt, mit BPS allein kann keine reale Anlage betrieben werden. Das Betriebssystem SVS kann auf virtuellen Maschinen arbeiten. Um eine effektive Arbeit des SVS auf virtuellen Maschinen zu sichern, wurden sowohl im SVS als auch im SVM umfangreiche Systemverbindungsmaßnahmen realisiert. Das SVM 3.3 ist eine aufwärtskompatible Weiterentwicklung des SVM 3.0/2/. Es besteht aus den Komponenten*

- Steuerprogramm (CP),
- dialogfähiges Programmier- und Textsystem (PTS),
- Fehleranalysensystem (PDAS),
- Dateifernübertragungssystem (RFTS) und
- Datenstationsvermittlung (TSW).

*Der Beginn der Auslieferung des SVM 3.3 ist für das 2. Halbjahr 1986 vorgesehen, nachdem im Rahmen einer Industrieprobung die stabile Funktionsfähigkeit gesichert wird. In den folgenden Punkten sollen die neuen Funktionen des SVM 3.3 erläutert werden.*

Weiterentwicklung des Steuerprogramms (CP)

Im CP werden umfangreiche Erweiterungen realisiert (Abb. 1). Die Komplexdatenstationsvermittlung, Dateipufferung, Fehlersuche und Systemverfolgung den in gesonderten Punkten behandelte SVM 3.3 unterstützt - außer den in aufgeführten Modellen - die Modelle ESER, Reihe 2, EC 1046, EC 106] 1066.

Es können jetzt 256 entfernte Bildschirme des Systems EC 7920 angesprochen werden. Die Ladbarkeit von **Prozessoren** (EC 8371, EC 8371.01. einer virtuellen Maschine unter Steuerung des SVM wird gesichert.

Eine wichtige Entwicklungslinie ist die Arbeit mit mehreren realen Anlagen bzw. auf die Arbeit innerhalb eines **Netzwerks** gerichtet. Bisher war es SVM möglich, mittels der Kommando RFTS Dateibertragungen mit Modulen der Datenfernverarbeitung zwischen Anlagen durchzuführen. Im SVM kann dafür als Übertragungsmittel der Kanal-Kanal-Adapter (KKA) genutzt und so eine schnelle Dateibertragung zwischen räumlich benachbarten Dialoganlagen realisiert werden. Für die Dialogarbeit mit verschiedenen DV-Anlagen eines Netzes wurde die Komponente Datenstationsvermittlung entwickelt (siehe TSW). Wenn zwei DV-Anlagen mit direkter Steuerung verbunden sind, kann die Signalübergabe auch bei der Arbeit dem Betriebssystem SVM vorgeordnet werden. Dazu ist es erforderlich, eine virtuelle Maschine für die Benutzung der Direktsignalsteuerung zu autorisieren. Diese kann dann die entsprechenden Lese- bzw. Schreibkommandos für den Signalaustausch mit der anderen Dialoganlage geben. In der Entwicklung des Betriebssystemkomplexes OC-7 EC ist die Unterstützung von DFV-Systemen vorgesehen. Die dafür erforderlichen Schlüsse werden im SVM 3.3 zur Verfügung gestellt. Es wird damit z. B. die Arbeitsweise ermöglicht, bei der ein Benutzer am Abonnementpunkt des Netzes an der virtuellen Konsole einer virtuellen Maschine arbeitet.

Viele Verbesserungen betreffen die Arbeit am Terminal. So kann mit dem Kommando SET RETRIEVE eine Programmfunktion belegt werden, daß bei jedem Dialog die zuletzt erfolgten Eingaben nach dem LIFO-Prinzip wieder in der Eingabe des Bildschirms erscheinen. Dabei ist gleichgültig, in welchem Modus die virtuelle Maschine gerade befindet. Die Benutzung einer sekundären Bedienung ist nicht mehr an eine Eintragung im Nutzerverzeichnis gebunden. Der Benutzer kann zu einem beliebigen Zeitpunkt entscheiden, daß er die Art

- Unterstützung neuer Modelle und Geräte
  - Direktsignalsteuerung zwischen zwei virtuellen Maschinen verschiedener realer Anlagen
  - Neue Komponente: Datenstationsvermittlung
  - Verbesserung zur Dateipufferung
  - Neue Hilfsmittel zur Fehlersuche
  - Neue Hilfsmittel zur Systemverfolgung
  - Erhöhung der Verfügbarkeit und Sicherung des Systems
- Abb. 1 Entwicklung im CP

einer virtuellen Maschine am Bildschirm einer anderen virtuellen Maschine mit verfolgen will. Mit dem Kommando QUERY FDISK kann man im SVM 3.3 die im Moment zur Verfügung stehenden freien Zylinder für temporäre Platten anzeigen lassen und ist so nicht mehr auf die bisherigen Problemmethoden angewiesen. Ein neues Kommando MTOR steuert die Nachrichtenübergabe zwischen virtuellen Maschinen, wobei eine Rückantwort erforderlich ist. So kann man am Bildschirm des Systembedieners alle noch nicht beantworteten (behandelten) Anforderungen in einem vorgebbaren Intervall immer wieder erscheinen lassen. Eine sehr unangenehme Begleiterscheinung von Systemabbrüchen ist der Verlust temporär angeforderter virtueller Platten. Im SVM 3.3 sind die Nutzer vor dem Verlust der temporär angeforderten Platten geschützt, wenn sie spätestens nach 15 Minuten das LOGON-Kommando wiederholen. Diese Zeitdauer kann bei der Generierung des SVM oder durch den Systembediener beim Warm-start auch verändert werden.

**Datenstationsvermittlung (TSW)**

Die Datenstationsvermittlung ist eine neue Komponente des SVM. Sie ermöglicht einem Terminalnutzer die Dialogarbeit mit mehreren DV-Anlagen, die über KKA oder BSC-Leitungen verbunden sind.

Wenn das Terminal an einer lokalen Anlage angeschlossen ist und der Nutzer mit einer Zielanlage arbeiten will, dann ist es erforderlich, daß sowohl die lokale Anlage, als auch alle Anlagen, die sich auf dem Weg bis zur Zielanlage befinden, mit SVM betrieben werden und jeweils eine spezielle virtuelle Maschine (TSW-

Maschine) aktiv ist. Die Zielanlage kann mit SVM oder mit einem Betriebssystem betrieben werden, das die Geräte EC 7921 oder EC 7924 unterstützt (siehe Abb. 2).

Diese Funktion dient also der Dialogarbeit mit verschiedenen Rechnern, aber nicht der Datenübertragung zwischen ihnen. Dafür ist RFTS zu verwenden. Das Initialisieren der Verbindung kann auf zwei Weisen erfolgen (siehe Abb. 3):

- nach dem Einschalten des Terminals wird das Kommando DIAL TSW1 gegeben
- am Terminal wurde mit der virtuellen Maschine VM1 des lokalen Rechners gearbeitet. Es wird das Kommando CONTSW TSW1 gegeben.

In beiden Fällen erscheint dann am Bildschirm ein Auswahlbild über alle realen Anlagen, mit denen das Terminal verbunden werden kann. Nachdem der Cursor auf den Namen der gewünschten Anlage gesetzt und die Eingabetaste gedrückt wurde, erscheint das Startbild der Zielanlage. Die Arbeit beginnt dann normal mit z. B. LOGON VM2. Wenn das Initialisieren von einer virtuellen Maschine des lokalen Rechners aus erfolgt, kann man mit Hilfe einer Programmfunktionstaste den jeweils aktuellen Bildschirminhalt bei der Arbeit mit VM2 in eine virtuelle Platte der virtuellen Maschine VM1 kopieren. Das ist die einzige Möglichkeit, mit Hilfe von TSW zwischen den realen Rechnern Daten auszutauschen. In Abb. 4 ist eine mögliche Zwei-Richtungsverbindung mit Hilfe von TSW dargestellt.

**Verbessern der Dateipufferung**

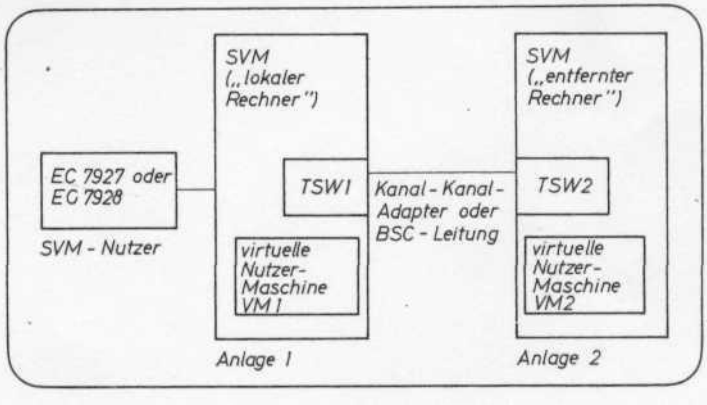
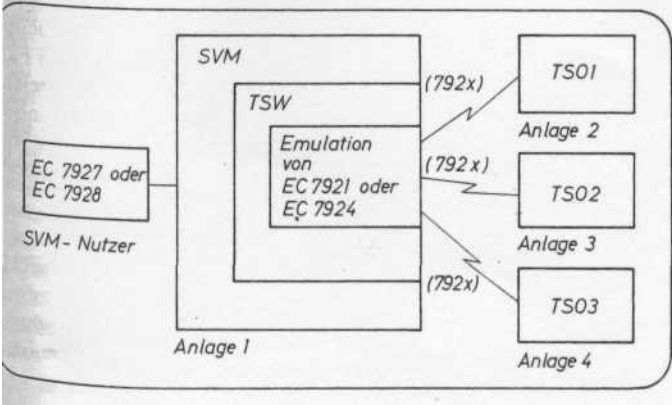
Die im SVM 3.3 vorgenommenen Verbesserungen zur Dateipufferung haben nur geringen Einfluß auf die Arbeit virtueller Maschinen mit der Kommandoklasse G. Sie sind jedoch sehr interessant für den Systembediener und den Systemverantwortlichen. Falls es notwendig wird, eine bestimmte Anzahl von Druck-Seiten zurückzusetzen, meist wegen gestörtem Papiervorschub, steht im SVM 3.0 das Kommando BACKSPAC zur Verfügung. Wenn in der SPOOL-Datei aber keine Kanalkommandos 'Vorschub zum Kanal 1' vorhanden sind, wird diese SPOOL-Datei auf jeden Fall bis zu ihrem

Anfang zurückgesetzt. Der neue Operand 'Line' ermöglicht zeilenweises Zurücksetzen; das wird in jedem Fall exakt ausgeführt, gleichgültig ob in der SPOOL-Datei Kanalkommandos 'Vorschub zum Kanal 1' vorhanden sind oder nicht. Das neue Kommando FWDSPACE arbeitet in umgekehrter Weise als das bekannte Kommando BACKSPAC. Es gibt verschiedene Gründe, von einer SPOOL-Datei eine bestimmte Anzahl Seiten beim Druck wegzulassen. Mit dem Kommando FWDSPACE 00E 99 TOF wird ein Weglassen der Seiten 1 bis 99 vom Anfang der SPOOL-Datei an veranlaßt, die SPOOL-Datei wird beginnend mit Seite 100 gedruckt.

Um eine Überlastung der SPOOL-Bereiche mit veralteten SPOOL-Dateien zu vermeiden, wird dem Systembediener bzw. Systemverantwortlichen die Möglichkeit geboten, alle SPOOL-Dateien zu löschen, die vor einem bestimmten Datum erzeugt worden sind. Beim Generieren kann mit dem Makro SYSSPCTL eine Aufbewahrungsfrist in Tagen angegeben werden. Beim Initialisieren des SVM werden automatisch alle SPOOL-Dateien gelöscht, die diese Aufbewahrungsfrist überschreiten. Ein Zusammenhang besteht zwischen der FWDSPACE-Möglichkeit und dem neuen Operanden REST im Kommando FLUSH cuu HOLD REST. Das Kommando FLUSH cuu HOLD REST gibt der Systembediener beispielsweise, wenn der SVM-Lauf beendet werden soll, aber im Augenblick eine große SPOOL-Datei im Druck befindlich ist, auf deren Beendigung nicht gewartet werden kann. Die Verbesserung des SVM 3.3 besteht darin, daß zu einem späteren Zeitpunkt die abgebrochene SPOOL-Datei nicht wieder von Anfang an gedruckt wird, sondern automatisch nur der bisher noch nicht gedruckte Teil der SPOOL-Datei. Das ist möglich, weil vom System die FWDSPACE-Möglichkeit automatisch angewendet wird. Auch bei Druck-Wiederholung von SPOOL-Dateien, deren Druck wegen Gerätefehlern abgebrochen wurde, wird automatisch nur der noch

Abb. 2 Zugriff zu einem Nicht-TSW-System über TSW

Abb. 3 Zugriff zu einem TSW-System über TSW



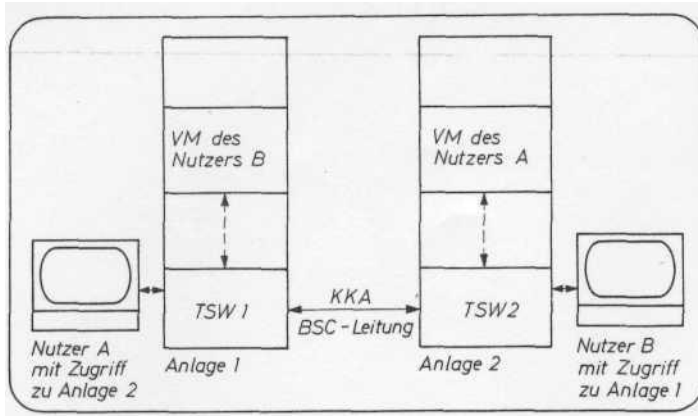


Abb.4 Zwei-Richtungs-Verbindung mit TSW

nicht gedruckte Teil der SPOOL-Datei ausgedruckt. Da die Möglichkeit besteht, daß der bisher ausgedruckte Teil einer solchen SPOOL-Datei verlorenght, kann mittels eines CHANGE-Kommandos vor dem Ausdruck der SPOOL-Datei die automatische REST-Ausgabe wieder rückgängig gemacht werden. Ein Nachteil der bisherigen SVM-Versionen besteht darin, daß bei System-Abbruch alle noch nicht abgeschlossenen SPOOL-Dateien verlorengehen. Hat beispielsweise ein Nutzer einer virtuellen Maschine für seine Druck-Ausgaben den Operanden CONT verwendet und druckt eine Anzahl kleinerer Dateien nacheinander, weil diese Arbeitsweise durch Auslassen der Separatoreseiten Papier spart, dann geht die SPOOL-Datei verloren, wenn der Nutzer vor dem System-Abbruch die Datei nicht mit dem Kommando SP PR CLOSE abgeschlossen hat. SVM 3.3 schließt bei System-Abbruch automatisch alle SPOOL-Dateien ab. Das führt dazu, daß auch die SPOOL-Dateien der virtuellen Bedieneinheiten den Zustand bis zum System-Abbruch wiedergeben. Wenn der Systembediener im SVM 3.3 das Kommando SHUTDOWN gibt, muß er vorher nicht für das Beenden aller virtuellen Maschinen sorgen, weil auch

bei SHUTDOWN alle SPOOL-Dateien automatisch abgeschlossen werden. Bei der neuen Spardruckfunktion des SVM 3.3 geht es um das effektive Nutzen von Drucker-Papier. Die effektivste Art ist das Drucken über den Falz hinweg. Diese Art ist aber nicht für alle Anwendungsfälle wünschenswert; um nicht über den Falz zu drucken, müssen vom Drucker Kanalkommandos 'Vorschub zum Kanal 1' realisiert werden. In einer Vielzahl von SPOOL-Dateien kommt ein solches Kanalkommando nach jeweils etwa 60 Druckzeilen. Es gibt Drucker, bei denen der Zeilenabstand variiert werden kann, zwecks Papiereinsparung wird oftmals der enge Zeilenabstand ausgewählt. In solchen Fällen wird die Absicht der Papiereinsparung durch den Vorschub zum Kanal 1 wieder durchkreuzt, das letzte Drittel der Druckseite bleibt leer. Wenn jedoch für den Drucker beispielsweise das Kommando START 00E SPARE YES LINCNT 85 gegeben wird, druckt dieser Drucker genau 85 Zeilen pro Seite und setzt dann auf den Anfang der nächsten Seite vor. Im SPOOL-File enthaltene Kanalkommandos 'Vorschub zum Kanal x' werden nicht realisiert; ihr Vorhandensein wird aber im Ausdruck vermerkt. Dem Systemprogrammierer wird im SVM 3.3 mit dem Kommando CPTRAP Möglichkeit geboten, ausgewählte Eintragungen oder alle Eintragungen der internen Verfolgungstabelle in eine SPOOL-Datei zu übertragen. Dadurch kann man auch größere Zeitabschnitte des SVM Laufs verfolgen, denn die SPOOL-Datei wird fortgeschrieben im Gegensatz Verfolgungstabelle im realen Speicher die vom Anfang an überschrieben wird, sobald sie gefüllt ist. Besonders viel Mühe wurde im SVM darauf verwendet, Systemblockierungen zu vermeiden, die dann auftreten, wenn die Plattenbereiche gefüllt sind, auf denen die SPOOL-Dateien stehen.

Diese Maßnahmen sind für alle virtuellen schienen interessant. Nicht immer ist sich ein Nutzer einer virtuellen Maschine bewusst, daß er große SPOOL-Dateien erzeugt. Deshalb kann beim Überschreiten der SPOOL-Datei-Größe durch eine bestimmte Anzahl Sätze der Systembediener und Bediener der virtuellen Maschine über diese Tatsache unterrichtet werden. Danach wird die SPOOL-Datei in die SPOOL-Klasse H umgesetzt. Größe der SPOOL-Datei und die SPOOL-Klasse kann bei der Generierung im Makro SYSSPCTL angegeben werden. Durch das Kommando QUERY SPOOL kann sich jeder Nutzer informieren, wie weit der SPOOL-Plattenbereich des Systems bereits gefüllt ist. Sobald der Füllungs-

grad 80 Prozent übersteigt, wird der Systembediener davon unterrichtet, mit er Maßnahmen ergreift, die Füllung des SPOOL-Plattenbereichs zu verringern.

Sobald der Füllungsgrad 90 Prozent übersteigt, wird die Weiterarbeit so virtuellen Maschinen verhindert, die gerade dabei sind, SPOOL-Dateien für virtuelle Drucker oder Stanzer zu erzeugen. Wenn es sich dabei um die virtuelle Maschine des Systembedieners handelt, wird ihre Weiterarbeit nicht behindert. SPOOL-Dateien für virtuelle Bedieneinheiten werden von diesem Moment an nicht mehr fortgeschrieben, die virtuelle Maschine erhält eine entsprechende Nachricht und kann ihre Arbeit fortsetzen, wenn sie auf die Fortschreibung der SPOOL-Datei für die Bedieneinheit verzichten will und wenn diese ihre einzige SPOOL-Aktivität war. Wenn die Weiterarbeit einer virtuellen Maschine verhindert wird, erhält sowohl der Systembediener als auch der Bediener der virtuellen Maschine eine Nachricht. Wenn der Füllungsgrad des SPOOL-Plattenbereichs wieder unter 85 Prozent absinkt, können alle in der Weiterarbeit behinderten virtuellen Maschinen automatisch weiterarbeiten.

Wenn der Bediener einer virtuellen Maschine nicht darauf warten will, daß die Arbeit seiner Maschine nach einer gewissen Zeit automatisch fortgesetzt wird, muß er seine SPOOL-Dateien für virtuelle Drucker oder Stanzer abschließen und gleichzeitig auch dafür sorgen, daß er von jetzt an keine SPOOL-Dateien mehr erzeugt.

### Neue Hilfsmittel zur Fehlersuche

Das TRACE-Kommando wird im SVM 3.3 unter dem Gesichtspunkt der gezielten Verfolgung bestimmter Aktivitäten der virtuellen Maschinen erweitert. Sie werden mit den TRACE-Operanden SVC, I/O, EXTERNAL, PRIV, SIO und CCW ausgewählt. Im SVM 3.3 entfällt die Einschränkung, daß die E/A-Operationen virtueller SPOOL-Geräte nicht mit TRACE verfolgt werden können. Durch den neuen Operanden RANGE, gefolgt von zwei Speicheradressen, wird ein Bereich festgelegt, innerhalb dessen Grenzen die TRACE-Anzeigen erfolgen. Dadurch ist es möglich, den Ablauf in einem Modul zu überwachen, ohne daß Verzweigungen in andere Module verfolgt werden.

Hilfsmittel zum Ermitteln fehlerhafter Programmabläufe, die bisher nur beim Test an der realen Anlage zur Verfügung standen, wie Überwachung der Veränderung bestimmter Speicherstellen, den Nutzern virtueller Maschinen zur Verfügung zu stellen, ist das Ziel des neuen PERTRACE-Kommandos. Das PERTRACE-Kommando benutzt die Einrichtung der Programmereignis-Registrierung (PER)/7/. Mit dem PERTRACE-Kommando wird eine völlig neue Qualität bei der Fehlersuche erreicht. Im Gegensatz zum Kommando **ADSTOP** kann beispielsweise der Aufruf eines an bestimmter Stelle des Speichers stehenden Befehls überwacht werden, ohne daß dieser Befehl zum Zeitpunkt der Kommando-Eingabe im Speicher stehen muß. PERTRACE kann den Aufruf eines oder mehrerer hintereinanderstehender Befehle überwachen. Dabei ist deren Befehlscode anzugeben. Besonders unbeliebte Fehler sind fehlerhafte Sprünge zu Speicherbereichen, an denen keine Befehle stehen, denn die im fehlerhaften Sprung vorausgehenden Befehle können nicht ermittelt werden. Dieses Problem löst PERTRACE, indem es an dem Befehl anhält, der in dem spezifizierten Speicherbereich hineinverzweigt. Zusätzlich werden die unmittelbar vorangegangenen fünf Verzweigungsbefehle angezeigt. Mit PERTRACE kann jede Veränderung eines bestimmten Speicherbereiches verfolgt werden, aber auch die Verfolgung der Veränderung auf einen ganz konkreten Inhalt kann für diesen Speicherbereich vorgenommen werden. So kann PERTRACE an dem Befehl anhalten, der auf die Speicherstelle '30000' bis

'30001' den Wert X'0000' einträgt. Mit PERTRACE kann die Veränderung bestimmter Bits an bestimmten Speicherstellen überwacht werden. Ebenso kann jede Veränderung eines bestimmten allgemeinen Registers verfolgt werden, aber auch die Verfolgung der Veränderung auf einen ganz konkreten Inhalt kann für dieses Register vorgenommen werden. PERTRACE kann an dem Befehl anhalten, der ins Register 8 den Wert X'FFFFFFF' stellt. Das Benutzen der Hilfsmittel PERTRACE und TRACE ist kombiniert möglich, das heißt, sie schließen sich nicht gegenseitig aus.

Hauptspeicheränderungen, die durch Kanalprogramme, das Unterbrechungssystem oder Zeitgeber verursacht werden, können von PERTRACE nicht überwacht werden. Durch die Verwendung des PERTRACE-Kommandos können auch längere Laufzeiten der Programme in der virtuellen Maschine auftreten. Die Verlangsamung durch die Überwachung ausgewählter Hauptspeicherbereiche ist jedoch relativ gering. Die Verlangsamung bei der Überwachung von Sprungbefehlen oder von Befehlen an einer bestimmten Speicheradresse bzw. von Befehlen mit einem spezifizierten Befehlscode kann das Mehrfache der normalen Programmlaufzeit erreichen. Das hat seine Ursache darin, daß bei der Überwachung der Verzweigung in einen bestimmten Programmbereich hinein jeder Sprungbefehl der virtuellen Maschine eine Programmunterbrechung auslöst, die vom CP behandelt werden muß. Also auch bei allen Sprungbefehlen, die nicht in den angegebenen Bereich hineinverzweigen. Bei der Überwachung eines Befehls auf einer bestimmten Speicheradresse erzeugt sogar jeder Befehl der virtuellen Maschine eine Programmunterbrechung, die vom CP behandelt werden

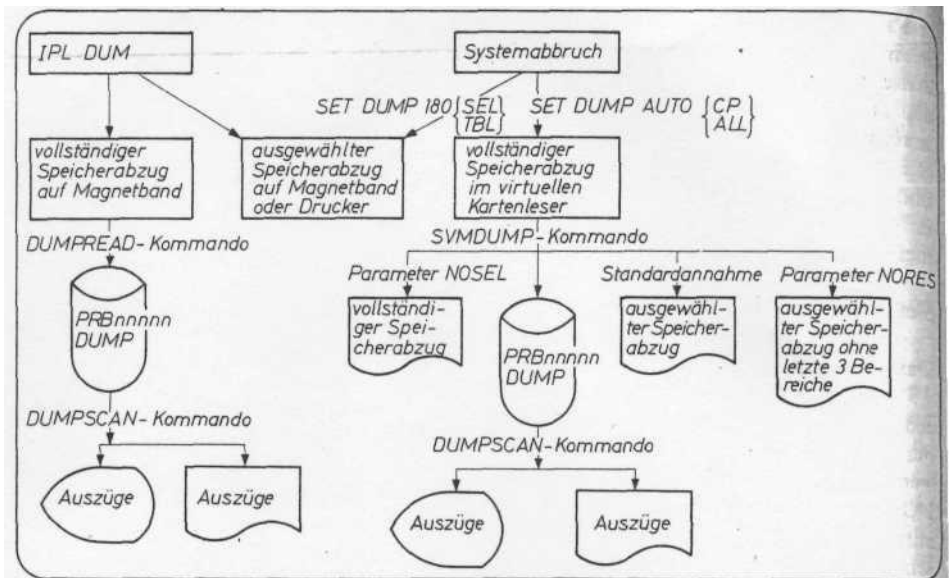
muß. Daraus ergibt sich die Empfehlung, daß vor der Benutzung des PERTRACE-Kommandos die virtuelle Maschine so nahe wie möglich an die Fehlerstelle herangebracht werden sollte und PERTRACE nicht unüberlegt für Abläufe aktiviert wird, für die es nicht nötig ist. Eine Nachricht informiert den Bediener einer virtuellen Maschine darüber, daß 10000 PER-Unterbrechungen für diese virtuelle Maschine verarbeitet worden sind, ohne daß eine Ausgabe auf der Bedieneinheit erfolgte.

Man kann Fehler sowohl durch die Verfolgung des Programmablaufs finden, als auch durch die Auswertung eines Speicherabzuges. Im SVM 3.3 werden neue Möglichkeiten der Bereitstellung von Speicherabzügen geliefert. Es gibt das systemunabhängige Programm IPL DUM auf der virtuellen Platte '190' des PTS. Man kann es von dort auf ein Magnetband oder auf Lochkarten bringen, damit es im Falle eines Fehlers von diesen Datenträgern geladen werden kann und einen Speicherabzug erzeugt. IPL DUM kann dann benutzt werden, wenn es dem System nicht möglich ist, von selbst einen Speicherabzug im Fall eines Fehlers zu produzieren. IPL DUM kann auch für andere Betriebssysteme als SVM benutzt werden und es ist auch für Speicherabzüge von virtuellen Maschinen einsetzbar. Im SVM 3.3 muß zwischen vollständigen Speicherabzügen und ausgewählten Speicherabzügen unterschieden werden. Ein vollständiger Speicherabzug hat immer das für PDAS notwendige Eingabeformat und kann weiterverarbeitet werden. Ein ausgewählter Speicherabzug hat immer das Format von Druckzeilen und kann nicht mittels PDAS weiterverarbeitet werden. Ein ausgewählter Speicherabzug gibt den Realspeicher in folgender Reihenfolge wieder:

- Seite Null des Realspeichers
- Interne Verfolgungstabelle
- DMKSYS und DMKRIO
- FREE-Bereiche
- Residenter Bereich des CP-Kerns
- fixierte Seiten des dynamischen Seitenbereichs
- V = R-Bereich und dynamischer Seitenbereich.

Die letzten drei Bereiche können durch Parameterangabe **TBL** im SET DUMP-Kommando oder Parameterangabe **NORES** im SVMDUMP-Kommando weggelassen werden.

Ein von IPL DUM erzeugter vollständiger Speicherabzug kann mit dem neuen PDAS-Kommando DUMPREAD in eine PTS-Datei verwandelt werden. Eine Übersicht über die neuen Möglichkeiten zum Erzeugen von Speicherabzügen gibt Abb. 5.



### Neue Hilfsmittel zur Systemverfolgung

Im SVM 3.3 werden dem Systemverantwortlichen zwei neue Hilfsmittel zur Verfügung gestellt, mit denen es möglich ist, Untersuchungen zur Effektivität der Arbeit des Systems vorzunehmen. In der virtuellen Platte '19E' des PTS wird das Programm **IPL DMON** bereitgestellt. DMON wird in einer virtuellen Maschine wie ein systemunabhängiges Programm abgearbeitet. Die Informationen von DMON sind besser und umfangreicher als die Informationen des CP-Kommandos **INDICATE**. INDICATE liefert lediglich Augenblicksinformationen, DMON liefert seine Kennwerte für den Augenblick, für die letzten 2,5 Minuten und für die gesamte Laufzeit seit dem SVM-Start.

Durch Kommandos kann ausgewählt werden, welche der folgenden Informationsgruppen angezeigt werden sollen:

- Informationen über Systemressourcen in ihrer Gesamtheit,
- Informationen über die Aktivität aller virtuellen Maschinen oder
- Informationen über die Aktivität aller peripheren Geräte.

In Abb. 6 wird der Bildschirminhalt mit Informationen über Systemressourcen in ihrer Gesamtheit dargestellt. Die zweite Zeile stellt den Prozentsatz der Prozessorzeit im Problemzustand dar. Unter der Spalte TOT steht der Mittelwert für die gesamte Zeit seit dem SVM-Start. Unter der Spalte EVR1 steht der Mittelwert für die letzten 2,5 Minuten und unter der Spalte LST der Mittelwert des letzten Meßintervalls. Die zweite Zeile wird fortgesetzt mit den Informationen über die E/A-Aktivität des Kanals 0. Unter der Spalte SIOS steht die Gesamtzahl aller SIO(F)-Befehle seit dem SVM-Start. Unter der Spalte EVR steht der Mittelwert der SIO-Befehle pro Sekunde seit dem SVM-Start. Unter der Spalte EVR1 steht der Mittelwert für die letzten 2,5 Minuten und unter der Spalte LST

CPU-PAGING	TOT	EVR1	LST	CHAN	SIOS	EVR	EVR1	LST	STORAGE-QUEUES-USER
Problem	2.9	5.1	6.3	0	606	0	0	0	REAL CORE 2,048K
PAGE WAIT	0.8	1.3	1.4	1	4,044	1	2	3	DYN CORE 664K
I/O WAIT	2.2	2.1	2.8						SHARE RES 0K
IDLE WAIT	35.5	2.6	0.0	S	4,650	2	2	3	USED STORAGE 29%
OVERHEAD	58.3	88.4	89.3						EVR1 LST
CPU BUSY	62	95	96						Q1 1.1 0
CPU IDLE	38	5	4						Q2 3.2 1
SVC	31,154	27.8	34.0						Q3 2.2 1
READS	769	0.4	0.6						E1/E2 1.1 0
WRITES	71	0.0	0.0						LOGGED ON 3
PAGING RATE	0	0							DIALLED 0
STEAL PAGE RATE	0	0							ACTIVE 0.9 1
PAGE LOADING OVRH	0	0							INACTIVE 2.0 2
REAL PAGE OVRH	2	2							ASLEEP 0.0 0
CALC PAGE OVRH	2	2							KIP 0.9 1

DMON SYSTEM STATUS CBM 3.3 06/06/85 15:31:48 AFTER IPL 0:34:00  
 INTERVAL 11 PAGE 1 OF 1 CPU 4341 NUMBR 035089 RUNNING

der Mittelwert des letzten Meßintervalls. Die letzte Eintragung der zweiten Zeile enthält die Größe des Realspeichers. Weitere Erläuterungen zu den DMON-Tabellen sind in 187 enthalten.

Ein weiteres Hilfsmittel zur Systemverfolgung ist das Programm Monitorstatistik (MCT). Bisher mußte der Anwender des SVM zur Analyse der Monitorstatistik eigene Programme entwickeln. MCT druckt Tabellen für jede der Monitorklassen, indem es die gemessenen Werte auswertet. Es werden auch Tabellen mit Mittelwerten ausgedruckt, die das Programm Monitorstatistik selbst errechnet hat. Mit Hilfe dieser Tabellen sind Aussagen zur Effektivität des Systems, zum Verwendungsumfang der Ressourcen der EDVA, zu charakteristischen Kennwerten der Nutzerarbeit und zu möglichen Engpässen im System möglich.

### Weiterentwicklung von PTS und PDAS

Die Weiterentwicklung der Komponente **PTS** gegenüber der Ausgabe 3.0 des SVM erfolgte in drei Richtungen. Einmal wurde die Struktur des PTS durch den Wegfall des benannten Systems **PTSZER** vereinfacht - das ergibt eine effektivere

Abb. 5 Möglichkeiten der Erzeugung von Speicherabzügen

Abb. 6 Beispiel für DMON

Arbeitsweise und eine einfachere Generierung des PTS. Die zweite Richtung besteht in der Erweiterung der Funktionen bereits bestehender Kommandos, so daß die Anwenderfreundlichkeit dieser Kommandos verbessert wurde. Die dritte Richtung besteht in der Einführung neuer Kommandos, womit der Funktionsumfang des PTS erweitert wurde.

Folgende Schwerpunkte seien hier genannt:

- Einführen der neuen Prozedursprache EXEC2
- Einführen neuer Kommandos beziehungsweise neuer Optionen bereits vorhandener Kommandos zur Unterstützung der Arbeit mit EXEC2-Prozeduren (EXEC10, RDR, LISTFILE, QUERY)
- Erweitern der Möglichkeiten der OS-Programmentwicklung durch Realisieren der Parameterübergabe an OS-Programme im OS-Format, Erweitern der Möglichkeiten des Kommandos DEBUG zur Testunterstützung sowie Unterstützung der erweiterten Zugriffsmethode VSAM2
- Verbessern der Arbeit mit virtuellen Platten durch Anwendung des Komman-

dos RESTORE zur Wiederherstellung virtueller PTS-Platten und der Option SKIP für die Kommandos ACCESS und FOR-MAT zum Übergehen defekter Blöcke

- erweiterte Anwendung des Kommandos TAPE um die Funktionen COPY und LIST zum Kopieren bzw. Ausdrucken von Magnetbändern
- Unterstützung der Übergabe von Zeichenketten, die länger als acht Zeichen sind, beim Aufruf von Kommandos (erweiterte Parameterliste). Die Komponente PDAS wurde in zweierlei Hinsicht weiterentwickelt. Einmal wurden die Kommandos zur Auswertung von Speicherabzügen des SVM-Steuerprogramms erweitert (DUMPSCAN, SVMDUMP). Zum anderen werden die Kommandos SVTAPE und SVSDUMP zum Aufbereiten von auf Magnetband vorliegenden SVS-Speicherabzügen als PTS-Datei sowie zu deren Analyse und Auswertung neu unterstützt. Folgende neue Funktionen werden erläutert.

**EXEC2-Prozessor**  
Zusätzlich zu den bereits vorhandenen Prozedurprozessoren EXEC und EXECE wurde der EXEC2-Prozessor in die Ausgabe 3.3 aufgenommen. Im wesentlichen erfolgte dies aus Kompatibilitätsgründen. Funktionell ist der EXEC2-Prozessor als Erweiterung des EXECE-Prozessors und auf dem Niveau EXECE-Prozessor einzuordnen. Es ist zu beachten, daß die Steuersprachen von EXEC, EXECE und EXEC2 nicht kompatibel sind. Es ist aber möglich, daß sich diese Prozeduren wechselseitig aufrufen.

- **Verarbeitung von Zeichenketten variabler Länge bis 255 Zeichen.** Damit werden wesentliche Möglichkeiten der Anwendung von EXEC2-Prozeduren eröffnet. Zusätzlich unterstützt wird dies durch die Einführung der erweiterten Parameterliste beim Aufruf von EXEC2-Prozeduren über die virtuelle Bedieneinheit, indem der EXEC2-Prozedur-Parameter mit einer Länge von mehr als 8 Byte übergeben werden kann

- **Unterprozeduren.** Es ist möglich, in einer EXEC2-Prozedur eine oder mehrere Unterprozeduren zu bilden, die als Funktion oder als Prozedur aufgerufen werden. An Unterprozeduren können Parameter übergeben werden. Diese Parameter ersetzen für die Dauer der Abarbeitung der Prozedur die Anzahl und Werte der speziellen Variablen &1,...,&n

- **Verbesserte Abarbeitungsfähigkeit.** EXEC2 benutzt einen internen Zeilenpeicher, in dem eine gewisse Anzahl von bereits interpretierten Zeilen gespeichert werden. Dies führt z.B. bei der Verarbeitung von Zyklen (&LOOP) zur Verkürzung der Abarbeitungszeit.
- **Abarbeiten von Zyklen.** Die Gestaltung von Zyklen durch die &LOOP-Anweisung wurde flexibler durch Einführung der WHILE- und UNTIL-Bedingungen zur Steuerung der Anzahl der Zyklenabarbeitung

- **Verarbeitung von Klein- und Großbuchstaben.** Mit der Steueranweisung &CASE M kann erreicht werden, daß alle durch die Steueranweisung &READ eingelesenen Zeilen unaufbereitet weiterverarbeitet werden. Das bringt z. B. Vorteile bei der Anwendung von EXEC2-Prozeduren zur Bearbeitung von TEPROS-Dateien (Textverarbeitung)

- **Steuern der Verarbeitung.** Durch die speziellen Variablen &COMLINE, &DEPTH, &FILEMODE, &FILENAME, &FILETYP, &FROM, &LINE und &LINK kann sowohl die Abarbeitung einer als auch verschachtelter EXEC2-Prozeduren flexibel gesteuert werden.

In der Ausgabe 3.0 des PTS konnten als XEDIT-Makros nur EXECE-Prozeduren verwendet werden. Ab Ausgabe 3.3 können sowohl EXECE- als auch EXEC2-Prozeduren als XEDIT-Makros verwendet und somit die Möglichkeiten beider Prozeduren genutzt werden. Im Gegensatz zum EXECE besitzt EXEC2 keine direkte Möglichkeit des Lesens/Schreibens aus/in PTS-Dateien. Durch Anwenden des neuen Kommandos EXECIO sowie der EXEC2-Steueranweisungen &READ und &STACK/&BEG-STACK kann dies indirekt verwirklicht werden. Das Kommando EXECIO bietet nämlich die Möglichkeit, Sätze aus PTS-Dateien in den Programmkellerspeicher zu schreiben und Sätze aus dem Programmkellerspeicher in PTS-Dateien zu schreiben.

Das Kommando EXECIO bietet dem Anwender darüber hinaus noch weitere Möglichkeiten der Arbeit mit den Programmkellerspeicher, die ebenfalls in EXEC2-Prozeduren genutzt werden können.

Das neue Kommando RDR (Bestimmen des Typs der am virtuellen Kartenleser anliegenden SPOOL-Dateien) sowie das Erweitern der Kommandos LISTFILE und QUERY um die Option STACK, bieten weitere Möglichkeiten, Informationen in den Programmkellerspeicher zu bringen und damit in Prozeduren zu verarbeiten.

## Erweiterung der OC-Programmentwicklung

In der Ausgabe 3.3 des SVM hat der PTS-Nutzer die Möglichkeit, die Parameterübergabe beim Aufruf von Programmen entsprechend den Konventionen im OC/EC durchführen zu lassen. Damit können solche Programme ungeändert unter Steuerung des PTS ausgeführt werden. Zur Unterstützung des Tests wurde das Kommando DEBUG um einige Unterkommandos erweitert bzw. es wurde "die Möglichkeit geschaffen, Haltepunkte in durch die OS-Makros LOAD, LINK und XCTL dynamisch nachgeladene Programme zu definieren (dynamische Haltepunkte).

Es sind dies folgende Unterkommandos:

C	- Anzeigen von Speicherbereichen im DKOI-Code	CSECT	- Definition eines Programmabschnittes, auf dessen Symbole durch andere Unterkommandos bezug genommen werden kann
DECHEX	- Konvertieren dezimaler in hexadezimale Zahlen	DBREAK	- Definieren dynamischer Haltepunkte
DDELETE	- Löschen dynamischer Haltepunkte	DCECT	- Definition eines Pseudoprogrammabschnittes, auf dessen Symbole durch andere Unterkommandos bezug genommen werden kann
LINES	- Rückübersetzung von Hauptspeicherinhalten in Assembleranweisungen	HEXDEC	- Konvertierung hexadezimaler in dezimale Zahlen
SYMBOLS	- Definiert, daß die Symbole einer vorher definierten CSECT oder DSECT verwendet werden sollen	XC	- Ausgabe von Hauptspeicherbereichen in hexadezimaler Form oder im DKOI-Code.

Es wird die erweiterte Zugriffsmethode VSAM2 unter PTS unterstützt. Damit erweitern sich die Anwendungsmöglichkeiten von VSAM bei der Unterstützung

- von Satznummernfolge-Dateien (RRDS)
- von Alternativ-Index (AIX)
- von Katalogrettebereichen (CRA)
- einer neuen Ausgangsroutine für die E/A-Fehlerbehandlung (EXECEPION-EXIT).

## Verbessern der Arbeit mit virtuellen Platten

Durch das neue Kommando **RESTORE** besteht die Möglichkeit, eine virtuelle PTS-Platte mit nicht lesbarem Plattenda-teilverzeichnis (MFD) teilweise oder vollständig wiederherzustellen. In den Ausgaben bis zum 3.0 mußte eine solche virtuelle Platte neu formatiert werden. Mit diesem Kommando kann auch eine mit dem Kommando ERASE gelöschte Datei wiederhergestellt werden. Die Option SKIP des Kommandos ACCESS bewirkt, daß im Falle eines nicht lesbaren Kettenblockes (FSTB) auf einer virtuellen Platte, die Platte mit Zugriff zu den restlichen Dateien zugewiesen wird. Die Option SKIP des Kommandos FOR-MAT bewirkt, daß defekte Spuren beim Formatieren übersprungen werden. In früheren Ausgaben wurde das Kommando beendet, so daß die hinter der de-

fehlen Spur liegenden Spuren nicht benutzt werden konnten.

## Erweitern des Kommandos TAPE

Es werden die Funktionen COPY und LIST neu unterstützt. Die Funktion COPY gestattet es, beliebig viele Dateien von einem Magnetband auf ein anderes zu kopieren.

Die Funktion LIST gestattet es, die Sätze von Dateien auf beliebigen Magnetbändern auf der virtuellen Bedieneinheit anzuzeigen oder auf dem virtuellen Drucker auszugeben.

Die Funktion SCAN wurde erweitert um die Anzeige der Satzanzahl, der Satzlänge, des Satzformats sowie das Erstellungsdatums.

## Erweiterte Parameterliste bei Kommandoeingabe über die virtuelle Bedieneinheit

Beim Aufruf von Kommandos über die virtuelle Bedieneinheit wurde bisher in allen Ausgaben des PTS an das Kommando als Parameterliste, nur die in 8 Byte lange Zeichenkette aufbereitete Eingabezeile übergeben. Ab Ausgabe 3.3 wird zusätzlich zu dieser Standardparameterliste eine erweiterte Parameterliste übergeben, die die unaufbereitete Eingabezeile beschreibt. Damit ergeben sich neben neuen Möglichkeiten für PTS-Kommandos (FILE-DEF, LISTFILE, OSRUN) auch solche für nutzergeschriebene Programme.

## SVS-Speicherabzugsauswertung durch PDAS

Durch das Kommando SVSTAPE können auf Magnetband durch das Dienstprogramm IMDSADMP hergestellte Speicherabzüge des SVS 7.1 eingelesen und als PTS-Datei aufbereitet werden. Durch das Kommando SVS DUMP können diese PTS-Dateien im Dialog analysiert werden. Dies wird ermöglicht durch die folgenden Unterkommandos:  
DISPLAY - Anzeigen von Speicherbereichen durch Angabe realer oder virtueller Adressen  
PRINT - Ausdrucken von Speicherbereichen bei Angabe realer oder virtueller Adressen  
CVT - Anzeigen des Steuerblockes CVT.

### Literatur:

/VLampenscherf, S., Schröder, A., Wagner, K.:  
System virtueller Maschinen (SVM/ES).  
rechentechnik/datenverarbeitung 18 (1981) 2,  
S.23 ///Heinecke, K., Schröder, A.:  
System virtueller Maschinen - SVM, Ausgabe  
3.0.  
edv-aspekte 3/84, S.5