

## Betriebssystem SVS-7.1

Dr. Karl-Heinz Männel  
VEB Robotron-Elektronik Dresden  
Günter Brusdeylins  
VEB Robotron-Projekt Dresden

Das Betriebssystem SVS-7.1 wurde als Bestandteil des Betriebssystemkomplexes OC-7.1 EC in Zusammenarbeit mit der UdSSR im Rahmen des ESER entwickelt und stellt eine aufwärtskompatible Weiterentwicklung des Betriebssystems SVS-7.0 dar /1/. Mit dem in /2/ vorgestellten Betriebssystem SVS-7.0 wurde dem Anwender ein leistungsfähiges Betriebssystem vorrangig für die Stapelverarbeitung bereitgestellt. Auf der Grundlage von Erkenntnissen aus der Anwendung des OC-6.1 EC und des SVS-7.0 wurde das SVS-7.1 umfassend weiterentwickelt. Diese Weiterentwicklung bietet dem Anwender die effektive Bereitstellung des Betriebssystems in Form von produktionsbereiten Systemen einschließlich der im Steuerprogramm integrierten Teilnehmerunterstützung TSO. Gegenüber SVS-7.0 wurde die Effektivität des SVS-7.1 weiter verbessert. Eine deutliche Effektivitätserhöhung wurde mit der Bereitstellung des TSO SVS 2.0 gegenüber dem TSO SVS im OC-6.1 EC erzielt. Mit dem SVS-7.1 werden dem Anwender weiterhin neue Komponenten, wie z. B. VSAM 2/7/ und eine große Anzahl neuer Funktionen, die sich entweder in der Anwendung als wichtig erwiesen haben oder die Anwendungsfreundlichkeit des Systems erhöhen, bereitgestellt. Die Geräteunterstützung wurde im SVS-7.1 erweitert. Im OC-7.1 EC wurden gleichzeitig die selbständig vertriebsfähigen Komponenten vom Typ 2 weiterentwickelt. Den Schwerpunkt bildete hierbei der Assembler 2 und Sortieren/Mischen 2. Die Effektivität des Assembler 2 wurde deutlich durch die Einführung vorgenerierter Makrodefinitionen erhöht. Das Verteilungssystem des SVS-7.1 einschließlich TSO befindet sich auf einem Magnetband. Bestandteile des OC-7.1 EC sind Weiterentwicklungen des SVM-3 und des BPS-7. Die Ausgabe 3.3 des SVM wird in /6/ vorgestellt. Diese Ausgabe des SVM ist für die Nutzung mit SVS-7.1 konzipiert. Die Systemverbindung zwischen SVM-3.3 und SVS-7.1 wurde verbessert und erweitert.

Mit dem BPS-7.1 wurde ein gegenüber BPS-7.0 aufwärtskompatibles Betriebssystem entwickelt. Hierbei wurden folgende Zielstellungen realisiert /8/:

- Erhöhen der Zuverlässigkeit
- Erhöhen der Kapazität der internen Pseudoplaten auf zwei Gigabyte
- Bereitstellen des Subsystems der Datenfernverarbeitung für BPS
- Effektiveres Bereitstellen des Betriebssystems
- Beseitigen von einigen bisherigen funktionellen Einschränkungen des BPS-7.0 gegenüber SVS im OC-6.1 EC.

Das BPS-7.1 besteht als Verteilungssystem aus vier Magnetbändern. Das Verteilungssystem schließt eine modifizierte Variante des SVM-3 ein. Das Betriebssystem SVS-7.1 wird in diesem Jahr zur Auslieferung kommen. Bis zu seiner Auslieferung wird das Betriebssystem umfassend unter Anwendungsbedingungen erprobt; die Erkenntnisse aus dieser Erprobung werden dann entweder vor der Auslieferung in das SVS-7.1 eingearbeitet oder bei der Entwicklung des SVS-7.1 berücksichtigt.

### Produktionsbereite Systeme - TKS

Das SVS-7.1 wird als produktionsreifes System abgekürzt TKS (turnkey distribution system) ausgeliefert. Das TKS zeichnet sich durch folgende Eigenschaften aus:

- einfache, schnelle und fehlerarme Systembereitstellung für den Anwender
  - kompakte Systembereitstellung; für das gesamte Betriebssystem ist **nur** ein Magnetband erforderlich
  - vereinfachte und direkte Systemwartung beim Anwender durch Einbringen von Programmwartungsbändern in Anwendungssysteme
  - hohe Systemflexibilität der Anwendungssysteme durch erhöhte Möglichkeiten der Steuerung des Systems über Parameter und durch die ständige Möglichkeit der Systemanpassung.
- Der bisherige Prozeß der Systemgenerierung entfällt. Für das angelieferte TKS und auch für Anwendungssysteme (Systeme, die bereits mit Anwenderprogrammen, Anhangsroutinen des Anwenders, Katalogeintragungen usw. vervollständigt sind) kann eine Systemanpassung durchgeführt werden, mit der folgende Funktionen realisiert werden:
- Anpassen der E/A-Konfiguration des Systems an die physische Konfiguration
  - Einfügen bzw. Streichen von Nutzer-SVCs
  - Einfügen bzw. Streichen von Nutzergerätetypnamen
  - Ändern des Namens der Systemresidenz
  - Ändern der Konsolkonfiguration
  - Ändern von Systemparametern in der Parameterbibliothek.

Diese Funktionen sind auch einzeln realisierbar. Wird bei der Systemanpassung

eine Änderung des Kerns des Steuerprogramms erforderlich, dann wird dieser Kern unter Nutzung des vorhandenen Kerns gebildet und ausgetauscht. Das Starterbetriebssystem entfällt dadurch vollständig. Der Aufwand für eine Systemanpassung beträgt bei umfassender Änderung der E/A-Konfiguration ca. 15 Min. (am EC 1055).

*Es stehen vier Varianten des TKS zur Verfügung:*

- Abzug eines 29-MBytes-WPS Produktionsfertigen Systems
- Abzug eines 29-MBytes-WPS TKS mit TSO
- Abzug eines 100-MBytes-WPS TKS
- Abzug eines 100-MBytes-WPS TKS mit TSO.

*Das jeweilige Verteilungsmagnetband hält:*

- systemunabhängige Dienstprogramm IBCDASDI und IBCDMPRS
- IEHDASDR-Abzug des jeweiligen TKS.

*Aus der TKS-Nutzung ergeben sich folgende Vorteile:*

- schnelle und sichere Übernahme des Systems durch den Anwender, die auf wendigen Systemgenerierungen entfallen
  - alle Funktionen und Komponenten (bis auf selbständig auslieferungsfähige Komponenten vom Typ zwei) sind im System vorhanden und damit entweder direkt oder durch Steuerung über Systemparameter nutzbar. Die Schnittstelle zwischen den Komponenten wurden deshalb stark vereinfacht. Die sich in Anwendung befindlichen Betriebssysteme werden dadurch standardisiert, die Vielfalt der sich in Anwendung befindlichen Systeme wird stark reduziert. Es wird erwartet, daß sich die Stabilität des Betriebssystems dadurch wesentlich erhöht
  - die Wartung und Betreuung des Betriebssystems beim Anwender wird sich durch die geplante Einführung von Programmwartungsbändern wesentlich einfacher gestalten. Bei der Aktualisierung wird das Prinzip des Modultauschs in Anwendungssystemen angewandt. Dieses Verfahren wird auch für Änderungen des Kerns verwendet.
- Das n-te Programmwartungsband umfaßt auch den Inhalt des (n-1)-ten, wodurch beim Anwender Aktualisierungsschritte ausgelassen werden können und der Prozeß der Systemwartung einfach und überschaubar ist.
- Das TKS führte zwangsläufig zur Veränderung der Technologie der Entwicklung. Die Komponentenbibliotheken blieben im wesentlichen erhalten, sie werden je doch nur noch beim Entwickler geführt. Die Struktur der Komponentenbibliotheken wurde so geändert, daß sie um Jobströme ergänzt wurden, durch deren Arbeitung das TKS beim Entwickler entsteht. Die Komponentenbibliotheken enthalten Makros in vorgenerierter Form. Diese Struktur stellt das sogenannte Basissystem dar.

## Assembler 2, Ausgabe 3.0

Bei Änderung eines Moduls in einer Bibliothek kann unter Verwendung des dazugehörigen Jobstroms des Basissystems mit Hilfe eines techno-logischen Programms der erforderliche Jobstrom für das Programmwartungsband gebildet werden.

Herstellen des TKS wird die Systemanpassung für die Standardkonfiguration vorgenommen, ebenso werden standardmäßige Nutzergerätenamen usw. ge-

bildet. Der für diese Standardkonfiguration verwendete Eingabestrom wird dem Anwender in der SYS1.SAMPLIB bereitgestellt. Der Eingabestrom kann gespeichert und mit dem Programm GENUC aktualisiert werden. In einem System mit TSO kann hierfür auch TSO verwendet werden. Der Eingangsstrom für eine Systemanpassung sollte ständig zur Dokumentierung und für nachfolgende Systemanpassungen zur Verfügung stehen.

Die Anweisungen zur Systemanpassung sind weitgehend kompatibel mit den Generierungsanweisungen des SVS-7.0.

*folgende Anweisungen werden verarbeitet:* CHANNEL, CONSOLE, IOCTRL, IODEVICE, UNITNAME, SVCTABLE und GENERATE.

*Geändert gegenüber SVS-7.0 wurden:* - CONSOLE wurde neu aufgenommen (dadurch werden die Konsolparameter beschrieben)

- GENERATE enthält keinen Parameter mehr.

Durch geringfügige Änderungen des Eingangsjobstroms für eine Systemgenerierung des SVS-7.0 können so die Eingangs-anweisungen für die Systemanpassung des SVS-7.1 gebildet werden. Durch genaues Zuschneiden der Konfiguration des Systems auf die tatsächliche vorhandene Konfiguration kann die Effektivität verbessert werden. Das TKS paßt sich teilweise auch der vorhandenen Konfiguration an. Beim Start des Systems mit 01F arbeitet das System automatisch im Schreibmaschinenmodus; von 010 und 011 im Displaymodus. Alle Zugriffsmethoden stehen im System zur Verfügung.

Im System sind weiterhin standardmäßig MMS, DTF, SMF, OLTEP und alle gerätespezifischen Moduln vorhanden.

Standardmäßig sind weiterhin Systemanschlüsse für TSO und MAMO vorhanden. Die Bestände der **Bibliothek** SYS1.PARMLIB erhalten für die Steuerung des Systems eine wesentlich größere Bedeutung.

Beim Start des Systems erkennt das TKS selbständig, auf welchem Modell es gestartet wurde, und wählt die Nutzung der erforderlichen Moduln der Fehlerbehandlung (Kanal- und Maschinenfehler) aus. Das Vorhandensein eines arbeitsbereiten MAMO wird ebenfalls beim Systemstart erkannt.

Mit der Weiterentwicklung des Steuerprogramms des Betriebssystems SVS-7 wurde der Assembler 2, der mit der Ausgabe 1.0 bereits für das OC-6.1 ES zur Verfügung stand, mit folgenden Zielstellungen weiterentwickelt:

- *Erhöhen der Effektivität des Assembler 2, das heißt Verkürzen der Übersetzungszeiten und Verringern des Ressourcenbedarfs*

Im einzelnen wurden folgende Maßnahmen realisiert:

- Erzeugen und Verarbeiten von vorgenerierten Makros
- SYSPRINT-Ausgabe mittels QSAM (bisher BSAM), wodurch Effektivitätsverbesserungen entstehen
- Optimieren der Prozeduren für die Übersetzung (Einführen ausreichend hoher Blockungsfaktoren)

- steht eine ausreichend große Region zur Verfügung, dann verbleibt die Arbeitsdatei im virtuellen Speicher. Andererseits wird die Arbeitsdatei als SPOOL-Datei angelegt

- *Funktionelle Erweiterungen*

Es wurden folgende funktionelle Erweiterungen realisiert:

- Überarbeiten der SYSTERM-Ausgabe zur Reduzierung der Druckausgaben und zur Fehlerausgabe bei Abbruch über SYSTERM

- Unterstützen des Mikrofilmausgabegerätes EC 7602 zur Ausgabe von Assemblerprotokollen

- Generieren der DD-Namen für Prozeduren.

Als Ergebnisse der Weiterentwicklung entstanden die Ausgabe 2.1 des Assembler 2, die gleichzeitig mit SVS-7.0 angeboten wurde /3/ und die Ausgabe 3.0, die gleichzeitig mit SVS-7.1 erscheint. Das Erzeugen und Verarbeiten der vorgenerierten Makrodefinitionen wird ab Ausgabe 3.0 des Assembler 2 unterstützt. Diese Ausgaben des Assembler 2 sind aufwärtskompatibel und mit OC-6.1 ES sowie mit den Betriebssystemen des **OC-7 ES nutzbar**. Ab SVS-7.1 werden alle Makrodefinitionen des Systems in vorgenerierter Form bereitgestellt, so daß für dieses Betriebssystem die Anwendung des Assembler 2, Ausgabe 3.0, erforderlich ist.

Bei den vorgenerierten Makrodefinitionen handelt es sich um das Ergebnis ei-

ner Teilübersetzung der Makrodefinitionen in Assemblersprache durch den Assembler 2. Die vorgenerierten Makrodefinitionen stellen eine für die weitere effektive Übersetzung geeignete Zwischenstufe dar. Das Erzeugen dieser Zwischenstufe erfordert den wesentlichsten Zeitannteil für die Übersetzung eines Makros. Die vorgenerierte Makrodefinition ist die für den Assembler verarbeitbare interne Darstellung aller Anweisungen einschließlich der Anweisungen zur bedingten Übersetzung und aller Modellanweisungen.

Die vorgenerierten Makrodefinitionen können auch für Makrodefinitionen des Anwenders genutzt werden, sie besitzen das Satzformat F bzw. FB bei einer Satzlänge von 80 Bytes.

Der Assembler 2, Ausgabe 3.0, erkennt vorgenerierte Makrodefinitionen und verarbeitet sie mit dem gleichen Ergebnis wie Makrodefinitionen in Quellform. In einer Makrobibliothek können sowohl Makros in der vorgenerierten als auch in der Quellform vorhanden sein. Durch COPY-Anweisungen einzufügende Bestände einer untergliederten Datei werden bei der Teilübersetzung (Erzeugen der vorgenerierten Makrodefinitionen) in die vorgenerierte Form der Makrodefinition überführt. COPY-Anweisungen für vorgenerierte Makrodefinitionen können in Quellmoduln verwendet werden.

In Tabelle 1 ist der mögliche Effektivitätsgewinn bei der Nutzung von vorgenerierten Makros an einem Beispiel dargestellt.

Da sich die Nutzung der vorgenerierten Makros nur auf die Verarbeitungszeiten des Assembler 2 (PPN) auswirkt, bleiben die erforderlichen SV- und PPO-Zeiten gleich.

Bei der Übersetzung von Makros reduziert sich bei Nutzung von vorgenerierten Makros die Verarbeitungszeit auf ca. 10 Prozent. Für die Übersetzung eines Programms in Assemblersprache bei Nutzung der im Betriebssystem bereitgestellten Makros wurde eine ZE-Zeit-Reduzierung von ca. 9 Prozent festgestellt.

Tab. 1 Übersetzungszeiten des Assembler 2, Ausgabe 3.0, bei Nutzung vorgenerierter Makros

	Übersetzen eines Makros		Übersetzen eines Programms	
	Makro nicht vorübersetzt	Makro vorübersetzt	Verwendete Makros nicht vorübersetzt	Verwendete Makros vorübersetzt
SV-Zeit [s]	2,8	2,9	8,6	8,6
PPO-Zeit [s]	0,7	0,7	0,7	0,7
PPN-Zeit [s]	8,8	0,9	120	108,2
ZE-Zeit [s]	12,3	4,5	129,3	117,5

## Teilnehmerunterstützung TSO SVS2.0

Mit der Ausgabe 7.1 des SVS steht dem Anwender TSO SVS 2.0 eine aufwärtskompatible Weiterentwicklung des TSO SVS 1.1 zur Verfügung. *Die Zielstellung für die umfassende Weiterentwicklung war:*

- deutliche Verbesserung des Antwortzeitverhaltens
  - Vereinfachen des Bereitstellens von TSO als Komponente vom Typ zwei
  - Realisieren neuer Funktionen.
- Eine allgemeine Übersicht zum TSO im OC-7 EC wurde bereits in 141 gegeben. Im Laufe der Entwicklung von TSO SVS 2.0 wurde die Konzeption gegenüber den in 14/ gemachten Angaben erweitert.

### Neue Funktionen im TSO SVS 2.0

*Die funktionelle Weiterentwicklung umfaßt:*

- neue oder erweiterte Funktionen der Kommandoverarbeiter
- Wegfall der bisher generierungsabhängigen Parameter
- Gewährleistung der sofortigen Nutzung von TSO im TKS
- wesentlich erweiterte Möglichkeiten für Kommandoprozeduren
- Nutzen von SPOOL-Dateien im Stapelbetrieb parallel zum TSO-Betrieb
- Bereitstellen von Standard-Kommandoprozeduren.

### Nutzung von TSO im TKS

Nach dem Rückspeichern eines produktionsbereiten Systems mit TSO ist dieses arbeitsfähig, wenn mindestens eine lokal angeschlossene Datenstation EC 7920 mit einer im TKS unterstützen Adresse vorhanden ist. Im TKS ist das erforderliche Nachrichten-Steuerprogramm enthalten. Zunächst ist nur der Teilnehmer TSOU-SER zur Arbeit mit TSO berechtigt. Dieser Teilnehmer verwendet die im TKS enthaltene LOGON-Prozedur TSL. Durch das ACCOUNT-Kommando können weitere Teilnehmer in die Teilnehmerattributdatei aufgenommen werden. Mit dieser Verfahrensweise entfallen die bisher nach der Generierung des Betriebssystems mit TSO notwendigen

Nachbereitungsarbeiten (z.B. Herstellung eines Nachrichtensteuerprogramms, Einbringen von Prozeduren usw.) vor dem ersten TSO-Start. Für die weiteren Anpassungsarbeiten des TKS und der Schaffung eines optimalen Anwendersystems kann jetzt sofort TSO verwendet werden.

## Neue oder erweiterte Funktionen der Kommandoverarbeiter

Für das Output-Kommando wurde eine neue Syntax eingeführt und zusätzlich der Operand KEEP aufgenommen. Damit ist es möglich, daß die Systemausgabedatei auch nach der Ausgabe durch das OUTPUT-Kommando erhalten bleibt. SPOOL-Dateien können mit dem OUTPUT-Kommando des TSO SVS 2.0 nicht verarbeitet werden. Es gibt aber die Möglichkeit, Systemausgabedateien einer bestimmten Ausgabeklasse auch bei aktivem SPOOL-System als temporäre sequentielle Dateien zu erzeugen. Diese können mit dem OUTPUT-Kommando in der bisherigen Weise verarbeitet werden. In den Kommandos **RUN**, **LINK** und **LOADGO** wird die Anwendung des Programmiersystems PASCAL neu unterstützt.

Das Kommando **TIME** liefert zusätzlich zu den bisherigen Informationen Belastungsgrößen des Systems. Damit ist für den Teilnehmer die zu erwartende Systemreaktion besser einschätzbar. Für den Editor wurden die neuen Unterkommandos **LISTFILE** und **UNNUM** neu entwickelt. Das Unterkommando **LISTFILE** dient der Anzeige des Inhaltes einer sequentiellen Datei oder eines Bestandes einer untergliederten Datei. Mit dem **UNNUM**-Unterkommando können Zeilennummern entfernt werden. Das Unterkommando **RENUM** wurde erweitert und erlaubt nun das Neunummerieren eines ausgewählten Nummernbereiches. Für **DELETE** und **DOWN** wurden Kurzformen eingeführt.

## Neue Möglichkeiten für Kommandoprozeduren

Kommandoprozeduren sind den Aufruf von Dienstprogrammen oder von Programmen des Anwenders zu vereinfachen. Bisher konnten außer der Anwendung symbolischer Parameter in einer Kommandoprozedur nur sequentielle Kommandofolgen verwendet werden. Jetzt werden eine Vielzahl von Prozeduren bereitgestellt, mit denen analog einer Programmiersprache Kommandoprozeduren gestaltet werden (siehe Tabellen 3, 4, 5).

## Maßnahmen zur Effektivitätssteigerung

- Zielstellung*
- Aus der Analyse des TSO-Betriebes in OC-6.1 EC, SVS, wurden folgende Maßnahmen zur Effektivitätssteigerung für TSO SVS 2.0 abgeleitet:
- Effektivieren des EDIT-Kommandoverarbeiters
  - verstärktes Nutzen der Daten SYS1.LPALIB
  - Reduzieren der Inanspruchnahme von Supervisordiensten
  - Verringern des ZE-Bedarfes in häufig durchlaufenen TSO-Steuerprogrammteilen
  - Einführen der Arbeitsweise ohne JOBAustausch.

### Effektivierung des EDIT-Kommandoverarbeiters

Auf Grund der besonderen Bedeutung des EDIT-Kommandos wurden folgende Maßnahmen zum Reduzieren der Verarbeitungszeit durchgeführt:

- Verringerung der für die Initialisierungsphase des EDIT-Kommandovorbereiters benötigten Zeit durch verbesserte Algorithmen
  - Reduzieren der E/A-Zugriffe beim sequentiellen Schreiben in die EDIT-Arbeitsdatei
  - durch Zusammenfassen von einzelnen Lademodul zu größeren Modulverbänden wurde die Verzweigung des Verbandes durch direkten Sprung realisiert.
- Wesentliche Teile des EDIT-Kommandos und der wichtigsten Unterkommandos wurden zu einem Modul verbunden.

### Reduzieren der Inanspruchnahme Supervisordiensten - Realisieren von Verbindungen

Die TSO-Serviceroutinen SCAN, PARSE, PUTLINE, GETLIN1 GET) wurden bisher mit der Makroanweisung **LINK** aufgerufen. Da diese Routinen häufig benutzt werden, wurden zum Reduzieren des ZE-Zeitbedarfs die Verzweigung zu den Serviceroutine durch einen einen direkten Sprung realisiert, ZU diesem Zweck wurden alle Kommandoverarbeiter überarbeitet. Dabei wurden ebenfalls die Makroanweisungen **LOAD** und **DELETE** für Serviceroutinen entfernt.

TSMCP	Startprozedur für das Standard-Nachrichten-Steuerprogramm
TSO	Startprozedur für das TSO-Steuerprogramm
TSTRACE	Startprozedur für das Protokollierungsprogramm
BRDR	Startprozedur für das Eingabeprogramm für TSO-Stapeljobs
TSL	LOGON-Prozedur
TSCOMP	Kommandoprozedur zum Verdichten untergliederter Dateien
TSCOPY	Kommandoprozedur zum Aufruf von
TSGENER	" " " " IEBCOPY.
TSLIST	" " " " IEBGENER
TSMAP	" " " " IEHLIST
TSPTPCH	" " " " IMBLIST
TSPROGM	" " " " IEBPTPCH
	" " " " IEHPRGM

Tab. 2 Standardprozeduren für TSO (in der Datei SVS 1. PROCLIB des TKS).

Die Kommandoprozeduren für den Aufruf der Dienstprogramme nutzen neue Funktionen des TSO SVS 2.0 für Kommandoprozeduren

Steueranweisungen	
ATTN	Ausgangsroutine zur Behandlung von Achtungsunterbrechungen während der Ausführung einer Kommando-prozedur
ERROR	Ausgangsroutine zur Fehlerbehandlung während der Ausführung einer Kommando-prozedur
EXIT	Beenden einer Kommando-prozedur
*RETURN	Rückgabe der Steuerung von einer Ausgangsroutine
CONTROL	Festlegen von Auswahlbedingungen für die Prozedur-ausführung
DATA	Beginn einer Gruppe von Kommandos und Unterkommandos
DO-END	Gruppe von Anweisungen
ENDDATA	Ende einer Gruppe von Kommandos und Unterkommandos
GLOBAL	Definieren von globalen Variablen
PROC	Festlegen symbolischer Parameter für die Kommando-prozedur
GOTO	Unbedingter Sprung innerhalb einer Kommando-prozedur
TERMIN	zeitweilige Übergabe der Steuerung an die Datenstation
WRITE [NR]	Ausgabe eines Textes an der Datenstation
Wertzuweisungen	
READ	Eingabe von Werten für symbolische Variable von der Datenstation
READDVAL	Aufteilung des Wertes von &SYSDVAL auf symbolische Parameter
SET	Wertzuweisung für eine symbolische Variable
Auswertung von Bedingungen	
DO-WHILE	Beginn einer Gruppe von Anweisungen
IF-THEN	Bedingter Sprung innerhalb einer Kommando-prozedur
WHEN	Auswerten eines Rückkehrcodes
Zugriff zu einer sequentiellen Datei (QSAM)	
CLOSEFILE	Abschließen einer Datei
GETFILE	Lesen eines Satzes
OPENFILE	Eröffnen einer Datei
PUTFILE	Schreiben eines Satzes

&SYSUID	Teilnehmerkennzeichen
&SYSPROC	Name der LOGON-Prozedur
&LASTCC	Letzter Rückkehrcode
&MAXCC	Höchster Rückkehrcode
&SYSDVAL	Prozedurname nach implizitem Aufruf
&SYSSCAN	Limit für Zeilenprüfung
&SYSDLM	Nummer der zur Rückgabe der Steuerung verwendeten Zeichenkette nach einer TERMIN-Anweisung
&SYSDVAL	Eingabezeile oder Parameter
&SYSDVAL	Anzeiger für verschachtelte Prozeduren
&SYSPCMD	Name des letzten Kommandos
&SYSSCMD	Name des letzten Unterkommandos
&SYSTIME	Tageszeit
&SYSDATE	Datum

&DATATYPE (ausdr)	Ermitteln des Datentyps des Ausdrucks:
NUM	- Ausdruck ist numerisch
CHAR	- Ausdruck enthält mindestens ein nichtnumerisches Zeichen
&EVAL (arithm-ausdr)	Berechnen des arithmetischen Ausdrucks
&LENGTH (ausdr)	Ermitteln der Länge vom Ergebnis des Ausdrucks
&STR (ausdr)	Festlegen einer Zeichenkette
&SUBSTR (arithm-ausdr1 :arithm-ausdr2 ,zeichenkette)	Festlegen einer Teilzeichenkette
arithm-ausdr1	- Anfangsposition der Teilzeichenkette
arithm-ausdr2	- Endposition der Teilzeichenkette
zeichenkette	- zu bearbeitende Zeichenkette

Tab. 3 Prozeduranweisungen für Kommando-prozeduren des TSO SVS 2.0 Tab. 4 System variable für Kommando-prozeduren des TSO SVS 2.0 Tab. 5 Standardfunktionen für Kommando-prozeduren des TSO SVS 2.0

In häufig genutzten TSO-Moduln wurde die Speicheranforderung geändert und dadurch die Anzahl der Makroanweisungen GETMAIN und FREEMAIN reduziert.

**Verstärktes Nutzen der SYS1.LPALIB** Alle parallel verwendbaren Kommando-verarbeiter wurden in die Datei SYS1.LPALIB aufgenommen. Das hat folgende Vorteile:

- der Systembedarf für das Programm-laden entfällt
  - zwei oder mehrere Teilnehmer, die den gleichen Modul benötigen, benutzen den gleichen Code, das heißt die Real-speicheranforderung wird reduziert
  - die am häufigsten verwendeten Moduln tendieren zum Verbleiben im Real-Speicher.
- Dadurch konnte auch der Zeitbedarf für das Starten von TSO wesentlich reduziert werden. Die Datei SYS1.CMDLIB wird dann nicht mehr benötigt. **Arbeitsweise ohne Jobaustausch** Die Ausnutzung des virtuellen Speichers (16 MBytes) erlaubt es, viele TSO-Regionen zu definieren. Damit ist die Möglichkeit gegeben, daß für jeden aktiven Teilnehmer eine TSO-Region vorhanden ist. Durch Angabe von **NOSWAPPING** in den Systemparametern für TSO SVS 2.0 wird die Arbeitsweise ohne Jobaustausch festgelegt. Damit wird jeder TSO-Region nur ein

Teilnehmer zugewiesen. Sind alle TSO-Regionen belegt, erfolgt ein Zurückweisen der Sitzungseröffnung für einen weiteren Teilnehmer. **Diese Arbeitsweise hat folgende Vorteile:**

- Wegfall des Aufwandes für die Organisation und die Durchführung des Jobaustausches, Damit verbunden ist ein starkes Reduzieren der Seitenwechsel, durch Wegfall des zwangsweisen Seitenwechsels für die Arbeitsmenge der TSO-Jobs
  - Verbessern des Antwortzeitverhaltens durch ständige Bereitschaft des TSO-Jobs zum Empfang einer ZE-Zeitscheibe (Wegfall der Hauptspeicher-Zeitscheiben)
  - Reduzieren des Zeitbedarfes für E/A-Operationen durch Nutzen von EXCPVR in den Zugriffsmethoden QSAM und BSAM. Bei der Arbeitsweise mit Jobaustausch ist wegen der Änderung der realen Adressen von fixierten Seiten beim Einräumen eines TSO-Jobs die Nutzung von Systemdiensten, die EXCPVR verwenden, nicht möglich
  - Verringern der Verarbeitungszeit durch Nutzen der Spurrpufferung für die Jobketten-datei beim Initialisieren einer Sitzung. In der Arbeitsweise ohne Jobaustausch entfällt die Spurrpufferung. Für Anlagen EC 1055, EC 1055 M und EC 1056 ist diese Arbeitsweise bevorzugt anzuwenden.
- Reicht der virtuelle Speicher nicht aus, um die Anzahl der für diese Arbeitsweise benötigten TSO-Regionen zu gewährleisten, ist wie bisher die Arbeitsweise mit Jobaustausch zu benutzen. **Verringerung des ZE-Bedarfs im TSO-Steuerprogramm** Zur Ermittlung der Benutzungshäufigkeit

von Befehlsfolgen im TSO-Steuerprogramm wurden Messungen unter typischen Anwendungsbedingungen durchgeführt. Auf dieser Grundlage wurden im TSO-Dispatcher, TSO-Interfaceprogramm und im TSO-Koordinator andere Verarbeitungsalgorithmen realisiert, die den ZE-Zeitbedarf reduzieren.

## Effektivitätsvergleich zwischen TSO SVS 2.0 und TSO SVS 1.0

### Meßverfahren

Eine wesentliche Größe zur Effektivitätsbeurteilung von TSO ist die Antwortzeit.

- **Antwortzeit** ist die Zeitspanne vom Auslösen eines Auftrages durch den Teilnehmer bis zum Empfang des Ergebnisses an der Datenstation (siehe Tabelle 6).

• **Übertragungszeit** ist die Zeitdauer der Nachrichtenübertragung zwischen Datenstation und Zentral-einheit. Sie ist von der Art des Anschlüssen abhängig. Bei entfernt angeschlossenen Datenstationen beeinflusst die Übertragungszeit wesentlich die Antwortzeit.

- **Reaktionszeit** ist im TSO-Betrieb mit Jobaustausch die Zeit, um den bereitgewordenen TSO-Job in den Hauptspeicher einzuräumen. Bei der Arbeitsweise ohne Jobaustausch ist die Reaktionszeit gleich Null.

• **Verarbeitungszeit** Diese Zeit kann sich bei der Arbeitsweise mit Jobaustausch aus mehreren Hauptzeitscheiben zusammensetzen und ist die Zeitdauer, die zum Erzeugen der Ergebnisse notwendig ist.

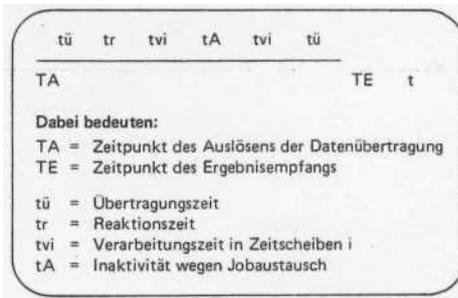
• Inaktivität wegen Jobaustausch  
Bei der Arbeitsweise mit Jobaustausch kommt es zum Jobaustausch, wenn in einer TSO-Region mehr als ein Teilnehmer arbeitet. Für den ausgeräumten Job entsteht dadurch eine Unterbrechung der Verarbeitung bis zum erneuten Einräumen in die TSO-Region. Bei der Arbeitsweise ohne Jobaustausch entfällt diese Inaktivität. *Der Effektivitätsvergleich* erfolgte durch Abarbeitung definierter Aufträge (Meßobjekte) in verschiedenen Betriebssystemen unter typischen Nutzungsbedingungen des jeweiligen Systems. Zur Zeitnahme diente das Kommando **TIME**. Damit wird die Startzeit der Verarbeitung und das Ende der Verarbeitungsphase ausgewiesen. Mit dieser Verfahrensweise wird der Einfluß der Übertragungszeit und der Reaktionszeit ausgeschaltet. Um eine Reproduzierbarkeit der Meßwerte zu erreichen, wurden die Messungen auf einer Rechenanlage durchgeführt, auf der keine weiteren Arbeiten liefen. Außer der Verarbeitungszeit wurden die während der Abarbeitung aufgetretenen SVC-Rufe, die Anzahl der Zugriffe zu den externen Geräten und der Verbrauch an ZE-Zeit erfaßt. Dieses Meßprogramm eliminiert seinen Eigenbedarf an ZE-Zeit. Die gemessene Verarbeitungszeit wurde um die Laufzeitverlängerung durch das Meßprogramm reduziert.

**Meßobjekte**

Nach der Befragung von Anwendern wurden zwei Aufträge gebildet, die TSO-Kommandos entsprechend der in der Praxis typischen Häufigkeitsverteilung enthalten.

Auf Grund der besonderen Bedeutung des EDIT-Kommandos wurde ein Auftrag zur Beurteilung der Effektivität dieses Kommandoverarbeiters gebildet. Dieser Auftrag enthält 1191 Kommandos und Unterkommandos.

Ein zweiter Auftrag umfaßt 381 Kommandos. Dieser Auftrag enthält die Kommandos **ALLOCATE, ATTRIB, FREE, RE-NAME, DELETE, LISTDS** und **CALL**. Diese Kommandos werden in typischen Anwendungsfällen häufig verwendet. Je Auftrag wurde eine Kommandoprozedur erstellt und dann in verschiedenen Systemen abgearbeitet. *Betrachtungen zur Effektivität des TSO SVS 2.0* Als Vergleichssystem für die Messungen



Tab. 6 Aufteilung der Antwortzeit

wurde das Betriebssystem OC-6.1 EC SVS 9. Modifikation mit TSO SVS 1.1 verwendet. Die Messungen erfolgten auf einer Anlage EC 1055.M mit einem 4 MByte großen Realspeicher und unter Verwendung von 29 MBytes Wechselplattenspeichern.

Zur Beurteilung der im TSO SVS 2.0 realisierten Maßnahmen zur Effektivitätssteigerung wurde zunächst eine Arbeitsfassung von TSO SVS 2.0 in eine Kopie des Vergleichssystems eingebracht. Diese Arbeitsfassung enthielt nicht die neuen Möglichkeiten für Kommandoprozeduren. Damit wurde erreicht, daß sich beide Systeme nur in der Komponente TSO unterschieden.

Für die TSO-Systemparameter wurde in beiden Fällen der Standardbestand IKJPRM00 der Datei SYSL.PARMLIB verwendet. In beiden Systemen wurden die beschriebenen Aufträge unter gleichen Bedingungen abgearbeitet. Weitere Messungen dienten dem Vergleich von SVS des OC-6.1 EC mit SVS 7.1. Hierzu einige Bemerkungen. Im TSO SVS 2.0 wurden die Möglichkei-

ten einer Kommandoprozedur wesentlich erweitert. Das hatte einen erhöhten Aufwand bei der Abarbeitung einer Kommandoprozedur gegenüber TSO SV1.1 zur Folge. Im TSO SVS 1.1 unterscheidet sich die Verarbeitung der durch eine Kommandoprozedur bereitgestellte Kommandos in keiner Weise von der Verarbeitung der durch manuelle Eingabe von der Datenstation eingegebenen Kommandos. Durch die neuen Möglichkeiten ergeben sich im TSO SVS 2.0 hierbei Unterschiede im Verarbeitungsaufwand, das heißt im TSO SVS 2.0 ist die Verarbeitung von Kommandos in einer Prozedur aufwendiger als die Verarbeitung der gleichen Kommandos bei manueller Eingabe. Für die manuelle Eingabe von Kommandos der häufigsten Art der Eingabe sind für TSO SVS 1.1 und TSO SVS 2.0 die Verarbeitungsbedingungen gleich. Um den erhöhten Aufwand für die Kommandoverarbeitung in einer Kommandoprozedur zu eliminieren und eine Aussage zur Verarbeitung für manuelle Eingabe zu erhalten, wurde im TSO SVS 2.0 für die Messung die Verarbeitung von Kommandoprozeduren analog TSO SVS 1.1 realisiert. Die damit gewonnenen Meßergebnisse sind für die Kommando Verarbeitung bei manueller Eingabe in TSO SVS 2.0 anwendbar. Aus den Ergebnissen der Vergleichsmessungen ist ersichtlich, daß für TSO im Betriebssystem SVS 7.1 eine erheblich Reduzierung der Verarbeitungszeit und damit der Antwortzeit erreicht wurde. In Tabelle 7 wurden die Meßergebnisse beider Aufträge addiert. Die damit gewonnenen Werte stellen ein für die TSO

	SVS 6.1 M9		SVS 7.1		
	TSO SVS 1.1	TSO SVS 2.0 manuell	rel. Aufwand zu TSO SVS 1.1 %	TSO SVS 2.0 Prozedur	rel. Aufwand zu TSO SVS 1.1 %
	1	2	3	4	5
Verarbeitungszeit [s]	877	421	48,0	489	55,8
SV-Zeit [s]	450,8	213,6	47,4	267,7	59,4
PPN-Zeit [s]	77,3	60,8	78,6	81,0	104,8
Anzahl SVC-Rufe:					
EXCP (0)		8341	64,4	9030	69,8
EXCPVR (114)	428	959	224,0	1057	246,9
EXIT (3)	46485	4590	9,9	4616	9,9
LINK (6)	6488	211	3,2	80	1,2
XCTL (7)	4816	500	10,4	496	10,3
LOAD (8)	4258	310	7,3	331	7,7
DELETE (9)	5672	240	4,2	261	4,6
FREEMAIN/GETMAIN (4)	5164	5162	100,0	6724	130,2
(5)	1314	1314	100,0	6108	464,8
(10)	51225	40603	79,3	54722	106,8
QMGR (90)	2487	2329	93,6	2750	110,6
STAX (96)	1144	1044	91,2	7511	656,6
ALLOC (99)	894	844	94,4	866	96,2
Anzahl SIO's:					
gesamt	26814	14195	52,9	15127	56,4
Jobkettendatei	7526	1346	17,9	1888	25,1
System (ohne PAGE, SPOOL)	8598	3136	36,5	3238	37,7
Meßobjekt	5060	5346	105,6	5614	110,9
Arbeitsplatte	5578	4244	76,1	4244	76,1
Anzahl TSO-Kommandos	1572	1572		1572	
Ø Verarb. Zeit/Kommando [s]	0,56	0,27	48,2	0,31	55,4

Tab. 7 Effektivitätsvergleich Auftrag 1, Auftrag 2 (Gesamtwerte)

Anwendung typisches Gesamtprofil dar. Fr dieses Aufgabenspektrum ergibt sich eine Reduzierung der Verarbeitungszeit von 48 Prozent für die Kommandoverarbeitung bei manueller Eingabe und von 56 % bei Anwendung von Kommandoprozeduren.

Die durchschnittliche Verarbeitungszeit pro Kommando oder Unterkommando reduziert sich von 0,56 Sekunden auf 0,27 bzw. 0,31 Sekunden.

## Weiterung des SVS-7.1 - Betrachtungen zur Effektivität

Die funktionellen Erweiterungen des SVS-7.1 dienen folgenden Zielstellungen:

- Erhöhen der Effektivität des Betriebssystems
- Verbessern der Anwenderfreundlichkeit
- Vervollständigen der Systemverbindung zum SVM-3
- Erhöhen der Zuverlässigkeit des Betriebssystems
- Unterstützen neuer Gerätetechnik.

## Funktionelle Erweiterungen des SVS-7.1 zur Erhöhung der Effektivität

Das Erhöhen der Effektivität des Betriebssystems SVS-7.1 wurde durch Reduzieren der benötigten Supervisorzeit und der Kanal-, Gerätesteuereinheits- und Gerätebelastung erzielt. Ausgehend von der Analyse der Zugriffe zur Jobkettendatei im SVS-7.0 wurden für das SVS-7.1 Maßnahmen zur Reduzierung der Zugriffe abgeleitet. Alle bisherigen Funktionen des Systemwiederanlaufs wurden beibehalten und die Kompatibilität der Nutzung der Jobkettendatei auf Anwenderprogrammiveau blieb erhalten.

Die Anzahl der physischen Zugriffe zur Jobkettendatei wurde durch Vermeidung von Lesezugriffen gesenkt. Für den Stapelbetrieb wurde eine Senkung auf 67 Prozent und für den Dialogbetrieb mit TSO eine Senkung auf 18 Prozent gemessen. Die Schreibzugriffe zur Jobkettendatei sind für die Funktionsfähigkeit des Systemwiederanlaufs erforderlich. Die Lesezugriffe konnten durch Einführen einer verdichteten Jobkettendatei in der SQA und durch Einführung eines Puffers für Steuerblöcke in der **LSQA** eingespart werden. Die Steuerblöcke in der LSQA werden mit der Datenträgerzuordnung angelegt. Das Aktualisieren dieser Steuerblöcke ist in der Regel erst bei Jobschrittende erforderlich, Zum Gewährleisten der Anwenderkompatibilität beim Nutzen der Jobkette war es erforderlich, eine EXCP-Simulation für Jobkettenzugriffe einzuführen.

Der Steuerprogrammaufwand wurde im Supervisor im SVS-7.1 weiter reduziert. Im SVS-7.1 erfolgt für Typ-1-SVCs, die unter Steuerung der Nutzersegmenttabelle gegeben werden, kein Laden der Systemsegmenttabelle mehr. Dadurch konnte der Steueraufwand für Typ-1-SVCs um 27 Prozent gesenkt werden. Häufig auftretende Typ-2-SVCs werden im SVS-7.1 als Typ-1-SVC gestartet. Sie laufen dadurch nicht unterbrechbar und wesentlich verkürzt ab. Werden in Ausnahmefällen Bedingungen festgestellt, die die Unterbrechbarkeit erfordern, so erfolgt der Übergang zum Typ 2. Der Steueraufwand für solche SVCs konnte im SVS-7.1 gegenüber SVS-7.0 auf 28 Prozent reduziert werden. Die Anzahl der EXITs wurde dadurch reduziert, daß alle EXITs, die im SVS-7.0 noch im SV-Zustand gegeben wurden, auf die Nutzung der Direktverbindung umgestellt wurden. Messungen zeigten ein Reduzieren der Anzahl der EXITs um 13 Prozent im SVS-7.1. Im SVS-7.1 wurde der Direktverbindungseingang von EXIT so geändert, daß bei aktiver TRACE-Funktion der SVC ausgelöst wird und damit in der Verfolgungstabelle auftritt. Dadurch vereinfacht sich die Fehlersuche.

Im SVS-7.1 wurde die jobschrittbezogene Wartezeitmessung durch Nutzen der TOD-Uhr effektiviert. Durch das Integrieren von TSO in das Steuerprogramm erhöhte sich der Aufwand im Supervisor, insgesamt wurde jedoch im SVS-7.1 eine Reduzierung gegenüber SVS-7.0 erreicht. Die BSAM/QSAM-Effektivität wurde im SVS-7.1 weiter erhöht. Es wurden folgende *standardmäßig zu kettende Kanalprogramme eingeführt*:

- Eingabe von Magnetplatte: 25
- Ausgabe auf Magnetplatte: 20
- Ein-, Ausgabe Magnetband: 28.

Beim Schreiben auf eine Magnetplatte wird die Anzahl der erforderlichen E/A-Operationen dadurch gesenkt, daß bei großen Blocklängen für das Beschreiben einer Spur nur noch eine Ausgabeoperation erforderlich ist.

Im SVS-7.1 wird beim Nutzen von TSO im Modus ohne Jobaustausch für Zugriffe über BSAM und QSAM **EXCPVR** verwendet; im Modus mit Jobaustausch **EXCP**.

Die standardmäßige Kettung erfolgt im SVS-7.1 bei QSAM-Nutzung auch im Falle der EXCP-Verwendung. Für das Strukturieren von BDAM-Dateien (BSAM-LOAD) wurde im SVS-7.1 **EXCPVR** eingeführt. In V=R-Bereichen wird im SVS-7.0 beim Nutzen von BSAM und QSAM ebenfalls EXCPVR verwendet. Die Steuerabläufe sind in diesem Fall aufwendiger als EXCP, insbesondere bei der Arbeit unter SVM. Deshalb wurde im SVS-7.1 die Arbeit von BSAM und QSAM in V=R-Bereichen auf EXCP umgestellt.

Im SVS-7.1 wurde SMF standardmäßig in das Betriebssystem eingeschlossen. Beim Systeminitialisieren kann SMF unterdrückt werden. Das Erfassen der SMF-Daten wird dann in allen Komponenten unterdrückt. Der Steueraufwand von SMF wurde reduziert mit Hilfe von SVCs und durch Optimierung der Standardpuffergröße. Messungen ergaben bei SMF-Nutzung von SVS-7.1 eine SV-Zeiteinsparung von etwa 10 Prozent gegenüber SVS-7.0.

Zum Erhöhen der Effektivität beim Nutzen des Dienstprogramms IEBCOPY wurde EXCPVR in V=V-Bereichen des SVS-7.1 und für TSO im Modus ohne Jobaustausch eingeführt. Messungen, die unter vergleichbaren Bedingungen zu 121 vorgenommen wurden, ergaben bei Nichtnutzung von SMF für die Abarbeitung des in III beschriebenen Meßjobstromes eine Supervisorzeiteinsparung von 12 Prozent und eine Einsparung an Problemprogrammzeit von 5 Prozent für SVS-7.1 gegenüber SVS-7.0. Die Zugriffe zur Jobkettendatei reduzierten sich im SVS-7.1 um etwa 30 Prozent. Es wurde eine Durchsatzerhöhung von 7,5 Prozent gemessen.

## Verbessern der Anwendungsfreundlichkeit

Im SVS-7.1 wurde die Möglichkeit geschaffen, über ein Kommando (CHNGDUMP) SYSABEND- und/oder SYSUDUMP-Speicherausdrucke zu unterdrücken, einzelne Ausdruckkomplexe hinzuzufügen oder einzelne Ausdruckkomplexe zu unterdrücken (Kern, Steuerblöcke, LSQA, LPA oder Rettbereich). Im OS/ES werden bei fehlerfreier Systemausgabe für einen Job alle Daten für diese Systemausgabe gelöscht. Nachträglich festgestellte Mängel der Ausgabe, z. B. Papierstau, schlechte Druckqualität u. ä., machten ein erneutes Abarbeiten der Jobs erforderlich. Das SVS-7.1 bietet die Möglichkeit, parallel zum Druck die Systemausgabe des OPR auf Magnetband zu speichern. Durch diese verfahrensweise wird ein papiersparender Nachdruck bei auftretenden Fehlern ermöglicht. Für jede auf den Magnetbändern gespeicherte Datei wird eine Eintragung in einer Verzeichnistabelle erzeugt. Diese Verzeichniseintragung enthält alle Angaben zur Identifizierung und originalgetreuen Wiederholung der Ausgabe. *In der Jobverwaltung wurden weiterhin folgende Funktionen neu bereitgestellt:*

- Möglichkeit des Anschlusses von Nutzeranhangsroutinen im OPR
- Unterstützen privater Prozedurbibliotheken in Jobströmen
- Unterstützen alternativer Zweige in Jobnetzen und des Jobnetz-wiederanlaufs
- das Systemausgabe- bzw. Systemeingabeprogramm kann in der OPEN-Phase abgebrochen werden, wodurch bei fehlerhaft eingegebenem Startkommando kein Systemneuladen erforderlich wird.

Die Druckausgabe von Dienstprogrammen wurde zur Papiereinsparung reduziert. Zur Erleichterung der Arbeit mit den systemunabhängigen Dienstprogrammen IBCDASDI und IBCDMTRS wurden Korrekturmöglichkeiten bei fehlerhafter Eingabe und die Möglichkeit der Eingabe von Steueranweisungen im Dialog geschaffen.

Literatur:

- IV*/Münch, W.: Das Betriebssystem OS/ES, edv-aspekte 3/84, S.2  
*I2I* Männel, K.-H.; Schneider, S.: Betriebssystem SVS 7.0. edv-aspekte 3/84, S. 10  
*/3*/Kretzschmar, R.: Assembler 2. edv-aspekte 3/84, S.25  
*/4*/ Brusdeylins, G.: Teilnehmerunterstützung TSO. edv-aspekte 3/84, S.27  
*/5*/ Heinecke, K.; Schröder, A.: System virtueller Maschinen - SVM, Ausgabe 3.0. edv-aspekte 3/84, S.5  
*/6*/ Heinecke, K.; Gerbing; Lilpop: SVM, Ausgabe 33  
*H*/Neumann, J.; Thiele, R.; Scheler, K.-D.: Erweiterung des Funktionsumfangs der Zugriffsmethode VSAM, edv-aspekte 3/84, S.20  
*/8*/Wetzel, B.: BPS - Ein Betriebssystem des ESER für Stapelbetrieb, edv-aspekte 3/84, S. 9