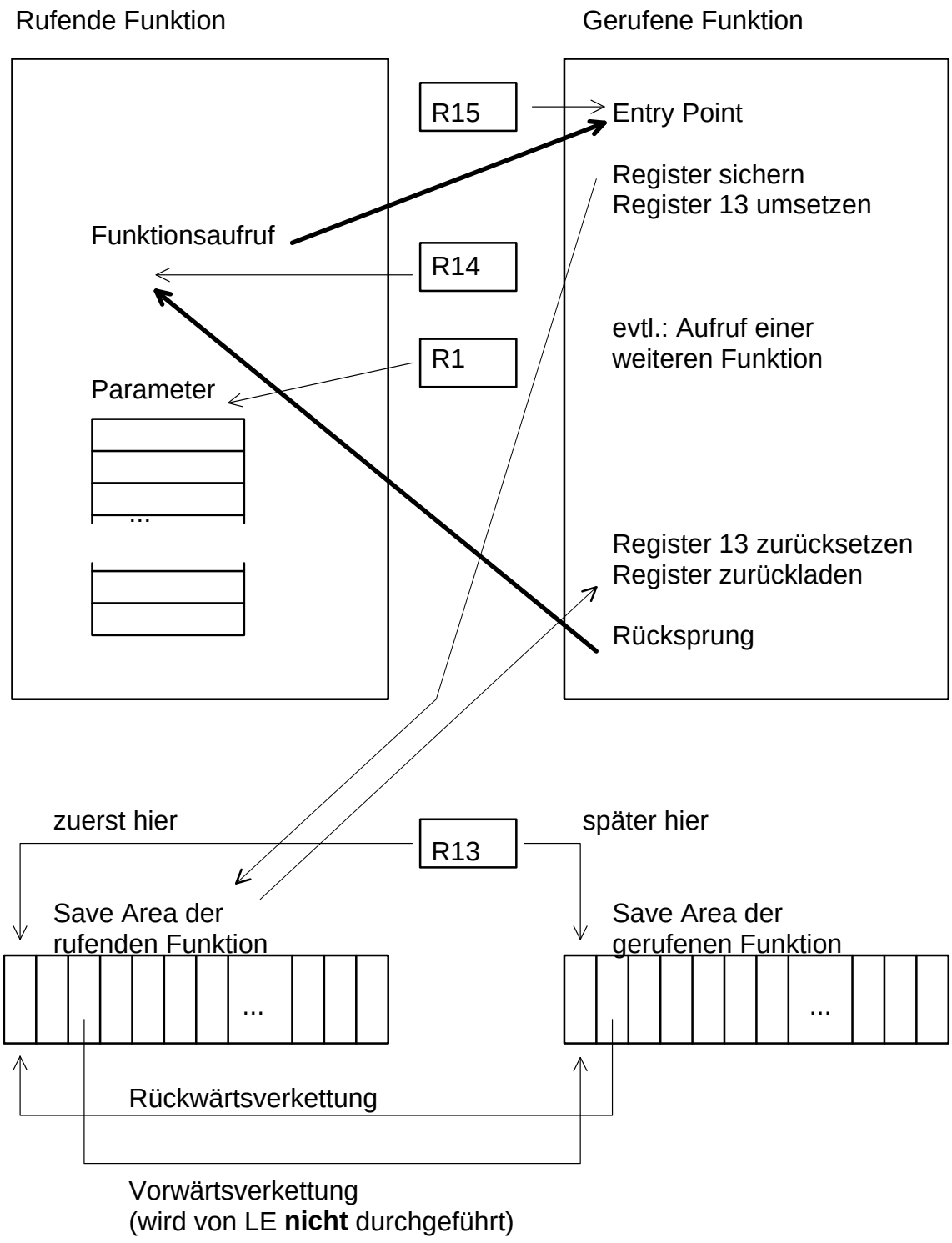
Kapitel 8

z/OS-Linkage-Konventionen, Save Area Trace

z/OS-Linkage-Konventionen



Erläuterungen:

Die Inhalte der 16 allgemeinen Register müssen beim Aufruf einer Unterfunktion **gesichert** und bei der Rückkehr wieder **rekonstruiert** werden.

Zu diesem Zweck stellt die **rufende** Funktion einen Sicherungsbereich, die sogenannte **Save Area**, zur Verfügung. Aufgabe der **gerufenen** Funktion ist es dann, sofort nach der Übergabe der Kontrolle die Registerinhalte in die Save Area der **rufenden** Funktion zu sichern. Die Register können dann in der Folge für die Zwecke der Unterfunktion frei verwendet werden.

Bei der Rückkehr rekonstruiert die **gerufene** Funktion die vorherigen Registerstände aus der Save Area der **rufenden** Funktion.

Die **Übergabe der Steuerung an eine Unterfunktion** läuft dabei in folgenden Schritten ab:

a) Die **rufende** Funktion sorgt dafür, dass das **Register 13** auf einen 72 Bytes langen Sicherungsbereich zeigt (die sogenannte **Save Area**).

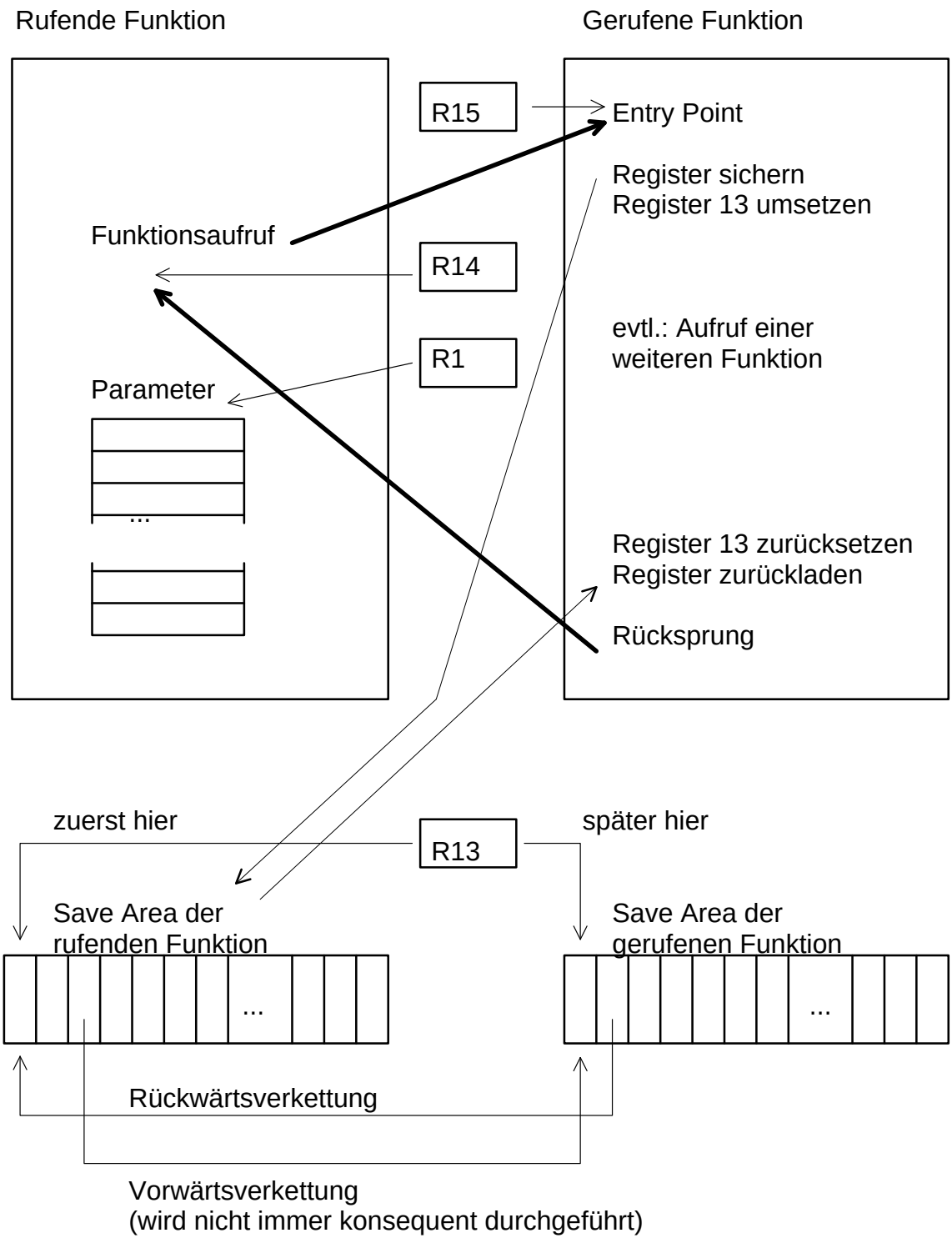
b) Die **rufende** Funktion sorgt weiterhin dafür, dass das **Register 15** den **Entry Point** der **aufzurufenden** Funktion und das **Register 1** ggf. einen Zeiger auf eine Liste von **Parametern** enthält.

c) Mit dem Maschinenbefehl BALR 14,15 bzw. BASR 14,15 (oder einem vergleichbaren Mechanismus des Betriebssystems) wird die Unterfunktion aufgerufen. Der BALR- bzw. BASR-Befehl sorgt dafür, dass die **Rückkehradresse** (d.h. die Adresse des auf den BALR- bzw. BASR-Befehl folgenden Befehls) in das **Register 14** eingetragen wird.

d) Die **gerufene** Funktion kopiert als erstes die Registerinhalte der **rufenden** Funktion in die Save Area.

e) Die **gerufene** Funktion benötigt selbst auch eine Save Area, um evtl. weitere Funktionen aufrufen zu können. Sie reserviert also Speicher für diese Save Area, verkettet die beiden Save Areas (siehe später) und stellt das **Register 13** auf die Anfangsadresse **ihrer Save Area**. Damit ist die Übergabe der Steuerung an die neue Funktion endgültig vollzogen.

z/OS-Linkage-Konventionen



Die **Rückgabe der Steuerung an die rufende Funktion** geschieht wie folgt:

a) Die **gerufene** Funktion stellt zunächst das **Register 13** wieder auf die **Save Area** der **rufenden** Funktion zurück.

b) Aus dieser Save Area **rekonstruiert** sie die Registerstände der **rufenden** Funktion (inklusive **Register 14**, das die **Rückkehradresse** enthält).

c) In das **Register 15** stellt die gerufene Funktion bei Bedarf einen **Returncode** ein (beziehungsweise bei C-Funktionen: das **Funktionsergebnis**, sofern es in ein Register passt).

d) Die gerufene Funktion springt an die im **Register 14** stehende Adresse und bewirkt somit den **Rücksprung**.

Übersicht über die Linkage-Register

Register 0 wird normalerweise beim Aufruf von Unterfunktionen nicht benutzt (aber beim Aufruf von Systemdiensten via SVC-Befehl).

Register 1 enthält beim Aufruf einer Unterfunktion die Anfangsadresse der **Parameter.**

Register 13 Dort muss die rufende Funktion die Anfangsadresse ihrer **Save Area** ablegen, damit die gerufene Funktion diese Save Area für die Sicherung der Registerstände der rufenden Funktion verwenden kann.

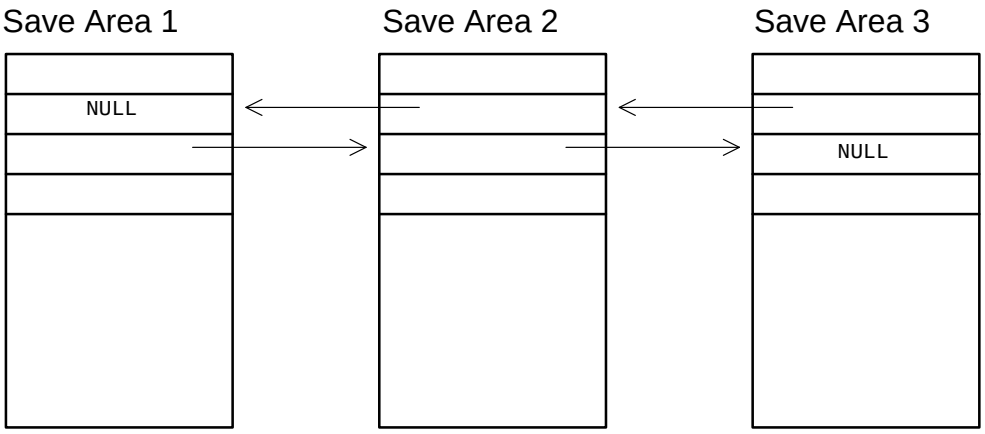
Register 14 enthält beim Aufruf die **Rückkehradresse** in die rufende Funktion. Sie muss von der gerufenen Funktion abgespeichert (sichergestellt) werden.

Register 15 enthält beim Aufruf der Unterfunktion den **Entry Point**, bei der Rückkehr den **Returncode** bzw. in C das **Funktionsergebnis**.

Der Aufbau der Save Area

0	Reserviert für LE (Language Environment)
4	Adresse der Save Area der rufenden Funktion (HSA)
8	Adresse der Save Area der gerufenen Funktion (LSA)
12	Register 14 (Rückkehradresse)
16	Register 15 (Entry Point)
20	Register 0
24	Register 1 (Parameter)
	...
68	Register 12

Verkettung der Save Areas:



Der Aufbau der Save Area

Die Save Area umfasst 18 Vollworte (72 Bytes), beginnt auf einer Wortgrenze und hat den nebenstehend beschriebenen Aufbau.

Bei C-Programmen wird die Save Area größer als 72 Bytes definiert. Sie enthält außer den hier angegebenen Feldern für die Registersicherung noch weitere Felder für Steuerungszwecke und außerdem **alle Automatic-Variablen**. Die Save Area wird bei C auch DSA genannt; DSA ist die Abkürzung für **Dynamic Save Area** bzw. **Dynamic Storage Area**.

Die Verkettung der Save Areas

Die Worte 2 und 3 in den Save Areas dienen dazu, die Save Areas untereinander zu verketten (SAVE AREA CHAIN). Das System hat auf diese Weise die Möglichkeit, im Fehlerfall im Dump Informationen über die Aufrufverschachtelung der Unterfunktionen auszugeben.

Wort 2 enthält die Adresse der Save Area des **Vorgängers** (der rufenden Funktion). In **Wort 3** trägt eine gerufene Unterfunktion die Adresse ihrer eigenen Save Area ein. Damit zeigt Wort 3 auf die Save Area des **Nachfolgers**.

Bemerkung:

Bei den von LE unterstützten Programmiersprachen fehlt oft die Vorwärtsverkettung über Wort 3. LE hält sich offenbar nicht konsequent an die z/OS-Linkage-Konventionen.

Das ist bedauerlich, da der SYSUDUMP im Save Area Trace die Kette der Save Areas nur dann vollständig anzeigt, wenn die Vorwärtsverkettung gesetzt ist. Da bei C-Hauptprogrammen gleich die erste Program Unit CEESTART die Verkettung nicht in der vorgeschriebenen Weise durchführt, wird im SYSUDUMP nur ein ganz kleiner Teil der Save-Area-Kette in aufbereiteter Form angedruckt. Die anderen Save Areas müssen in den ausgedruckten Hauptspeicherbereichen des USER SUBPOOL STORAGE gesucht werden.

```
CEE3DMP V1 R4.0: Condition processing resulted in the unhandled condition.        04/14/05 10:26:33 AM            Page:    1

Information for enclave main

  Information for thread 8000000000000000

  Traceback:
    DSA Addr  Program Unit  PU Addr   PU Offset  Entry         E Addr    E  Offset   Statement  Load Mod  Service  Status
    000264E0  CEEHDSP       0614C300  +00003C72  CEEHDSP       0614C300  +00003C72              CEEPLPKA  UQ76741  Call
    000263C8                40702260  +000002C6  sa9996        40702260  +000002C6              DMPX04H            Exception
    000261F8                40701030  +00000394  main          40701030  +00000394              DMPX04H            Call
    000260E0                06063716  +000000B4  EDCZMINV      06063716  +000000B4              CEEEV003           Call
    00026018  CEEBBEXT      0000E868  +000001A6  CEEBBEXT      0000E868  +000001A6              CEEBINIT  DRIVER8  Call


...


  Parameters, Registers, and Variables for Active Routines:

    CEEHDSP (DSA address 000264E0):
      UPSTACK DSA
      Saved Registers:
        GPR0..... 00000000  GPR1..... 000268FC  GPR2..... 00016038  GPR3..... 00000003
        GPR4..... 0001F778  GPR5..... 00000080  GPR6..... 00026D90  GPR7..... 00000008
        GPR8..... 8614FB48  GPR9..... 000274DF  GPR10.... 06150B4C  GPR11.... 0614C300
        GPR12.... 00020A50  GPR13.... 000264E0  GPR14.... 800170E6  GPR15.... 8616B4C8

    sa9996 (DSA address 000263C8):
      UPSTACK DSA
      Saved Registers:
        GPR0..... 0000000D  GPR1..... 000264B9  GPR2..... 00026360  GPR3..... 4070229A
        GPR4..... 00026498  GPR5..... 000264BC  GPR6..... 00000004  GPR7..... 00000025
        GPR8..... 00000032  GPR9..... 0002647C  GPR10.... FFFFF008  GPR11.... 00000005
        GPR12.... 00020A50  GPR13.... 000263C8  GPR14.... 0002647C  GPR15.... 00000000

    main (DSA address 000261F8):
      UPSTACK DSA
      Saved Registers:
        GPR0..... 4071F0BD  GPR1..... 00026290  GPR2..... 40701650  GPR3..... 4070106A
        GPR4..... 4071F154  GPR5..... 00000050  GPR6..... 0000000F  GPR7..... 40701755
        GPR8..... 000262D8  GPR9..... 40701788  GPR10.... 407017CA  GPR11.... 00026360
        GPR12.... 00020A50  GPR13.... 000261F8  GPR14.... C07013C6  GPR15.... 40702260

...
```

Der Save Area Trace im CEEDUMP

Die Save Areas werden im CEEDUMP mehrfach ausgegeben, und zwar, da die Vorwärtsverkettung von LE nicht durchgeführt wird, generell in Rückwärtsrichtung.

Zunächst erscheinen die Save Areas (= DSAs) im sogenannten Traceback; zusätzlich zu den DSA-Adressen werden jeweils die Entry Points und deren Adressen angegeben, sowie die dazugehörigen Funktionsnamen.

Etwas später erscheinen dann die Save Areas noch einmal, unter der Überschrift "Parameters, Registers, and Variables for Active Routines".

Beachten Sie bitte, dass die Entry Points der Funktionen (Register 15 zum Zeitpunkt des Funktionsaufrufs) jeweils in der Save Area des Rufers gespeichert werden; so steht beispielsweise der Entry Point der Funktion sa9996 in der DSA von main.

Dasselbe gilt für die Parameter: der Stand von Register 1 zum Zeitpunkt des Aufrufs der Funktion sa9996 (Basisadresse der Parameter für sa9996) steht in der Save Area von main.

Demgegenüber stimmt die Zuordnung für die auto-Variablen: die Basisadresse für die auto-Variablen von sa9996 ist gleichzeitig die DSA-Adresse von sa9996.

```
Control Blocks for Active Routines:

  DSA for CEEHDSP: 000264E0
    +000000  FLAGS.... 0808      member... CEE1      BKC..... 000263C8  FWC..... 00029078  R14..... 800170E6
    +000010  R15..... 8616B4C8  R0..... 00000000  R1..... 000268FC  R2..... 00016038  R3..... 00000003
    +000024  R4..... 0001F778  R5..... 00000080  R6..... 00026D90  R7..... 00000008  R8..... 8614FB48
    +000038  R9..... 000274DF  R10..... 06150B4C  R11..... 0614C300  R12..... 00020A50  reserved. 00046030
    +00004C  NAB..... 00029078  PNAB..... 0601BB7C  reserved. 800264A8  404084B8  00000000  00000005
    +000064  reserved. 00000000  reserved. 00000000  MODE..... 8614FF74  reserved. 4071F29C  00000004
    +000078  reserved. 00000001  reserved. 00000000

  DSA for sa9996: 000263C8
    +000000  FLAGS.... 1000      member... 0000      BKC..... 000261F8  FWC..... 00000000  R14..... C07023A6
    +000010  R15..... 0602E8E0  R0..... 0000000D  R1..... 00026460  R2..... 00026360  R3..... 4070229A
    +000024  R4..... 00026498  R5..... 000264BC  R6..... 000264BD  R7..... 00000025  R8..... 00000032
    +000038  R9..... 00026470  R10..... 407025C8  R11..... 00026360  R12..... 00000000  reserved. 00046030
    +00004C  NAB..... 000264E0  PNAB..... 00000000  reserved. 00000000  00000000  00000000  00000000
    +000064  reserved. 00000000  reserved. 00000000  MODE..... 00000000  reserved. 00000000  00000000
    +000078  reserved. 00000000  reserved. 00000000

  DSA for main: 000261F8
    +000000  FLAGS.... 1000      member... 0000      BKC..... 000260E0  FWC..... 00000000  R14..... C07013C6
    +000010  R15..... 40702260  R0..... 4071F0BD  R1..... 00026290  R2..... 40701650  R3..... 4070106A
    +000024  R4..... 4071F154  R5..... 00000050  R6..... 0000000F  R7..... 40701755  R8..... 000262D8
    +000038  R9..... 40701788  R10..... 407017CA  R11..... 00026360  R12..... 00020A50  reserved. 00046030
    +00004C  NAB..... 000263C8  PNAB..... 00000000  reserved. 00000030  00020910  00000000  06216E70
    +000064  reserved. 00020A50  reserved. 00000000  MODE..... 000262C0  reserved. 00000000  00000000
    +000078  reserved. 00000000  reserved. 00000000


Storage for Active Routines:

  DSA frame: 000263C8
    +000000 000263C8  10000000 000261F8 00000000 C07023A6  0602E8E0 0000000D 00026460 00026360  !....../8.......w..Y........-...-!
    +000020 000263E8  4070229A 00026498 000264BC 000264BD  00000025 00000032 00026470 407025C8  ! ......q..................... ..H!
    +000040 00026408  00026360 00000000 00046030 000264E0  00000000 00000000 00000000 00000000  !...-......-.....................!
    +000060 00026428  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000  !................................!
    +000080 00026448  00000000 00000000 00000000 00000000  00000000 00000000 00026470 407025C8  !............................... ..H!
    +0000A0 00026468  41300000 00000000 40F34BF0 F0F0009C  00026294 4DF36C5D 00F00040 85DAB806  !........ 3.000.....m(3%).0. e...!
    +0000C0 00026488  05DA6E38 00026470 00026510 00026460  C1E2D2D6 40C6C9D5 C1D5C3C5 40C24BE5  !..>.............-ASKO FINANCE B.V!
    +0000E0 000264A8  4B40C4D4 60C1D5D3 4B40F1F9 F8F64DF9  F35D40D6 D6000000 00000000 00000000  !. DM-ANL. 1986(93) OO...........!
    +000100 000264C8  00000010 00000002 0000000D 407025C8  00000001 00026480 0808CEE1 000263C8  !............ ..H..............H!

  DSA frame: 000261F8
    +000000 000261F8  10000000 000260E0 00000000 C07013C6  40702260 4071F0BD 00026290 40701650  !......-........F ..- .0..... ..&!
    +000020 00026218  4070106A 4071F154 00000050 0000000F  40701755 000262D8 40701788 407017CA  ! ... .1....&.... ......Q ..h ...!
    +000040 00026238  00026360 00020A50 00046030 000263C8  00000000 00000030 00020910 00000000  !...-...&..-....H...............!

    ...
```

Der Save Area Trace im CEEDUMP (2)

Die Save Areas werden noch zwei weitere Male im CEEDUMP ausgegeben, zunächst unter der Überschrift "Control Blocks for Active Routines".

Bei dieser Darstellung werden die einzelnen Teilfelder der Save Areas aufbereitet angezeigt, d.h. mit ihrer Bezeichnung. Man erkennt auch die Rückwärts- und Vorwärtspointer sowie die Felder der DSA, die hinter der Save Area des Betriebssystems liegen (ab Offset 72 = X'48').

Zum Schluss werden die Save Areas noch ein weiteres Mal angezeigt; diesmal ohne Erläuterungen (nur hexadezimal), dafür aber in voller Länge. Diese Darstellung dient dazu, automatic-Variablen zu suchen. Leider sind hier die Funktionsnamen nicht mit angegeben, aber aufgrund der DSA-Adressen lassen sich die Bereiche leicht den einzelnen Funktionen zuordnen.

```
SAVE AREA TRACE

PROCEEDING FORWARD FROM TCBFSA

INTERRUPT AT C070252A

PROCEEDING BACK VIA REG 13

NAME=UNKNOWN
WAS ENTERED VIA CALL         AT EP C
SA   000263C8  WD1 10000000   HSA 000261F8   LSA 00000000   RET C07023A6   EPA 0602E8E0   R0  0000000D
               R1  00026460   R2  00026360   R3  4070229A   R4  00026498   R5  000264BC   R6  000264BD
               R7  00000025   R8  00000032   R9  00026470   R10 407025C8   R11 00026360   R12 00000000

NAME=DMPX04H
WAS ENTERED VIA CALL         AT EP C
SA   000261F8  WD1 10000000   HSA 000260E0   LSA 00000000   RET C07013C6   EPA 40702260   R0  4071F0BD
               R1  00026290   R2  40701650   R3  4070106A   R4  4071F154   R5  00000050   R6  0000000F
               R7  40701755   R8  000262D8   R9  40701788   R10 407017CA   R11 00026360   R12 40713028



000213C0 00000000 00000000 00000000 00006F50    00026018 C07028CE 8000E868 7D000012   *.............?&..-.„.....Y.'...*
000213E0 40713190 00000000 00000000 00000000    00000000 00000000 00016038 40702238   * ........................-. ...*
00021400 008CF868 00000000 C0702802 00020A50    00046030 00026018 00026018 00000000   *..8....„.....&..-...-...-.....*
00021420 00000000 00000000 00000000 00000000    00000000 00000000 00000000 00000000   *................................*
00021440 00000000 00000000 00000000 00000000    C3C5C5E3 D4C5D4D3 00010028 00000000   *................CEETMEML........*

...

00026000 E2E3D2E4 00021194 00021194 00020000    00000000 00000000 00000000 000213C8   *STKU...m...m...................H*
00026020 000264A0 8000EA10 06063716 7D000012    00026098 40702768 00000002 8000E94C   *.............'.....-q .........Z<*
00026040 0001F778 4070223C 40702770 00000030    00000008 8606370A 8000E868 00020A50   *..7. ... ...........f.....Y....&*
00026060 00046030 000260E0 00000000 00000000    00000000 00000000 00000000 00000000   *..-...-™.......................*
00026080 00000000 00000000 00000000 00000000    00000000 00000000 0000EB1C 4070223C   *............................ ...*
000260A0 000260D0 0001F8B0 000260C4 000260C8    00000000 00000000 00000000 00000000   *..-•..8...-D..-H...............*
000260C0 00000000 40713190 00000000 00000000    00000001 00000000 00000000 00000007   *.... ..........................*
000260E0 10000000 00026018 00000001 800215C0    C0701030 000261F8 40713190 860637CA   *......-.......„„...../8 ...f...*
00026100 00000002 8000E94C 0001F778 4070223C    40702770 00000030 80000000 8606370A   *......Z<..7. ... ...........f...*
00026120 8000E868 00000000 00046030 000261F8    000260E0 000260E4 000260E8 000260F0   *..Y.......-.../8..-™..-U..-Y..-0*
00026140 000260EC 000260F4 00000000 00000001    00000001 00000000 00000000 00000000   *..-...-4.......................*
00026160 00000000 00000000 00000000 00000000    10000000 00026018 00026218 8001BDB2   *.....................-.........*
00026180 06216E70 00000000 00026208 00026100    05D627CA 00026124 40702638 05D630EA   *..>.........../..O..../. ....O..*
000261A0 00000000 00000030 00020910 00000001    86169888 00020A50 00046030 00026218   *................f.qh...&..-.....*
```

Der Save Area Trace im SYSUDUMP

Im SYSUDUMP werden die Save Areas aufbereitet ausgegeben entsprechend der durch die Vorwärts- und Rückwärtszeiger festgelegten Reihenfolge.

Zunächst wird die Kette der Save Areas in **Vorwärtsrichtung** ausgegeben, d.h. ausgehend vom Hauptprogramm. Dies wird durch die Überschrift "PROCEEDING FORWARD FROM TCBFSA" angezeigt; TCBFSA ist das Feld im TCB, das die Adresse der ersten Save Area (FSA = First Save Area) enthält.

Die Ausgabe der Vorwärtskette erfolgt jedoch nur, solange die Kette bzgl. der Vorwärts- und Rückwärtspointer konsistent ist (d.h., wenn Save Area A per Vorwärtspointer auf Save Area B zeigt, dann zeigt Save Area B per Rückwärtspointer auf Save Area A). Falls diese Bedingung nicht mehr erfüllt ist, wird die Ausgabe der Vorwärtskette abgebrochen (ggf. Meldung: "INVALID BACK CHAIN").

Da diese Bedingung bei C-Hauptprogrammen gleich in der ersten Program Unit (CEESTART) nicht erfüllt ist, wird die Vorwärtskette der Save Areas in der Regel nicht angezeigt.

Zur **Rückwärtsrichtung**:

Das Register 13 zeigt, wie bereits besprochen, immer auf die aktive Save Area. Nach der Ausgabe der Vorwärtskette wird ggf. die Kette ausgehend vom Register 13 rückwärts ausgegeben. Dabei werden jedoch nur zwei Save Areas aufbereitet und angedruckt. Die Rückwärtskette wird durch die Überschrift "PROCEEDING BACK VIA REG 13" gekennzeichnet.

Die fehlenden Save Areas können anhand der meistens intakten Rückwärtspointer im USER SUBPOOL STORAGE (hinterer Teil des Dumps) gefunden werden.

```
SAVE AREA TRACE

PROCEEDING FORWARD FROM TCBFSA

INTERRUPT AT C070252A

PROCEEDING BACK VIA REG 13

NAME=UNKNOWN
WAS ENTERED VIA CALL         AT EP C
SA   000263C8  WD1 10000000   HSA 000261F8   LSA 00000000   RET C07023A6   EPA 0602E8E0   R0  0000000D
               R1  00026460   R2  00026360   R3  4070229A   R4  00026498   R5  000264BC   R6  000264BD
               R7  00000025   R8  00000032   R9  00026470   R10 407025C8   R11 00026360   R12 00000000

NAME=DMPX04H
WAS ENTERED VIA CALL         AT EP C
SA   000261F8  WD1 10000000   HSA 000260E0   LSA 00000000   RET C07013C6   EPA 40702260   R0  4071F0BD
               R1  00026290   R2  40701650   R3  4070106A   R4  4071F154   R5  00000050   R6  0000000F
               R7  40701755   R8  000262D8   R9  40701788   R10 407017CA   R11 00026360   R12 40713028

000213C0 00000000 00000000 00000000 00006F50    00026018 C07028CE 8000E868 7D000012   *..............?&..-.„.....Y.'...*
000213E0 40713190 00000000 00000000 00000000    00000000 00000000 00016038 40702238   * ..........................-. ...*
00021400 008CF868 00000000 C0702802 00020A50    00046030 00026018 00026018 00000000   *..8.....„......&..-...-...-.....*
00021420 00000000 00000000 00000000 00000000    00000000 00000000 00000000 00000000   *................................*
00021440 00000000 00000000 00000000 00000000    C3C5C5E3 D4C5D4D3 00010028 00000000   *................CEETMEML........*

...

00026000 E2E3D2E4 00021194 00021194 00020000    00000000 00000000 00000000 000213C8   *STKU...m...m...................H*
00026020 000264A0 8000EA10 06063716 7D000012    00026098 40702768 00000002 8000E94C   *.............'.....-q .........Z<*
00026040 0001F778 4070223C 40702770 00000030    00000008 8606370A 8000E868 00020A50   *..7. ... ...........f.....Y....&*
00026060 00046030 000260E0 00000000 00000000    00000000 00000000 00000000 00000000   *..-...-™.......................*
00026080 00000000 00000000 00000000 00000000    00000000 00000000 0000EB1C 4070223C   *............................ ...*
000260A0 000260D0 0001F8B0 000260C4 000260C8    00000000 00000000 00000000 00000000   *..-•..8...-D..-H...............*
000260C0 00000000 40713190 00000000 00000000    00000001 00000000 00000000 00000007   *.... ..........................*
000260E0 10000000 00026018 00000001 800215C0    C0701030 000261F8 40713190 860637CA   *.....-........„„...../8 ...f...*
00026100 00000002 8000E94C 0001F778 4070223C    40702770 00000030 80000000 8606370A   *......Z<..7. ... ...........f...*
00026120 8000E868 00000000 00046030 000261F8    000260E0 000260E4 000260E8 000260F0   *..Y.......-.../8..-™..-U..-Y..-0*
00026140 000260EC 000260F4 00000000 00000001    00000001 00000000 00000000 00000000   *..-...-4.......................*
00026160 00000000 00000000 00000000 00000000    10000000 00026018 00026218 8001BDB2   *.....................-.........*
00026180 06216E70 00000000 00026208 00026100    05D627CA 00026124 40702638 05D630EA   *..>......./..O..../. ....O..*
000261A0 00000000 00000030 00020910 00000001    86169888 00020A50 00046030 00026218   *................f.qh...&..-.....*
```

Der Save Area Trace im SYSUDUMP (2)

Die einzelnen Teilfelder der Save Areas werden wie folgt beschriftet:

SA	Adresse der jeweiligen Save Area
WD1	Wort 1 (für PL/1 reserviert)
HSA	Higher Save Area (Rückwärtspointer)
LSA	Lower Save Area (Vorwärtspointer)
RET	Rückkehradresse bzw. Register 14
EPA	Entry-Point-Adresse bzw. Register 15
R0	Register 0
	usw.

Die erste Save Area nach "PROCEEDING BACK VIA REG 13" ist die Save Area, auf die Register 13 zeigt, also die DSA der aktiven Funktion. Die Anfangsadresse dieser Save Area ist gleichzeitig die **Basisadresse für die auto-Variablen** der aktiven Funktion.

Ansonsten ist vor allem die **nächste** Save Area interessant, also die der rufenden Funktion. Das liegt daran, dass die Registerstände zum Zeitpunkt des Aufrufs der aktiven Funktion dort gespeichert werden. Man kann dort den **Entry Point der aktiven Funktion** finden (**EPA**), die **Rückkehradresse**, also die Position innerhalb der rufenden Funktion, zu der die aktive Funktion zurückkehren würde (**RET**), und die **Anfangsadresse der Parameter**, die die rufende Funktion an die aktive Funktion übergeben hat (**R1**).

Die betreffenden Werte aus dem nebenstehenden Beispiel:

`4070252A`	Interrupt-Adresse bzw. PSW
`000263C8`	Anfangsadresse der DSA der aktiven Funktion
`40702260`	Entry Point (EPA) der aktiven Funktion
`407013C6`	Rückkehradresse (RET) zur rufenden Funktion
`00026290`	Anfangsadresse der Parameter (R1)

Weitere Werte:

`0602E8E0`	Entry Point (EPA) der von der aktiven Funktion zuletzt gerufenen Funktion
`407023A6`	Rückkehradresse (RET) zur aktiven Funktion, also Position des Funktionsaufrufs dieser Funktion innerhalb der aktiven Funktion

Zu beachten: das jeweils erste Bit ist nicht Bestandteil der Adresse, deshalb ist ggf. 8 von der ersten Hexziffer der Adresse abzuziehen.

```
SAVE AREA TRACE

PROCEEDING FORWARD FROM TCBFSA

INTERRUPT AT C070252A

PROCEEDING BACK VIA REG 13

NAME=UNKNOWN
WAS ENTERED VIA CALL         AT EP C
SA   000263C8  WD1 10000000   HSA 000261F8   LSA 00000000   RET C07023A6   EPA 0602E8E0   R0  0000000D
               R1  00026460   R2  00026360   R3  4070229A   R4  00026498   R5  000264BC   R6  000264BD
               R7  00000025   R8  00000032   R9  00026470   R10 407025C8   R11 00026360   R12 00000000

NAME=DMPX04H
WAS ENTERED VIA CALL         AT EP C
SA   000261F8  WD1 10000000   HSA 000260E0   LSA 00000000   RET C07013C6   EPA 40702260   R0  4071F0BD
               R1  00026290   R2  40701650   R3  4070106A   R4  4071F154   R5  00000050   R6  0000000F
               R7  40701755   R8  000262D8   R9  40701788   R10 407017CA   R11 00026360   R12 40713028




000260E0 10000000 00026018 00000001 800215C0    C0701030 000261F8 40713190 860637CA   *......-........„„...../8 ...f...*
00026100 00000002 8000E94C 0001F778 4070223C    40702770 00000030 80000000 8606370A   *......Z<..7. ... ...........f...*
00026120 8000E868 00000000 00046030 000261F8    000260E0 000260E4 000260E8 000260F0   *..Y.......-.../8..-™..-U..-Y..-0*
00026140 000260EC 000260F4 00000000 00000001    00000001 00000000 00000000 00000000   *..-....-4.....................*
00026160 00000000 00000000 00000000 00000000    10000000 00026018 00026218 8001BDB2   *.....................-.........*
00026180 06216E70 00000000 00026208 00026100    05D627CA 00026124 40702638 05D630EA   *..>......./..O..../. ....O..*
000261A0 00000000 00000030 00020910 00000001    86169888 00020A50 00046030 00026218   *.............f.qh...&..-.....*

...

40702240 40702768 00000000 F2F0F0F5 F0F4F1F3    F0F8F1F5 F5F9F0F1 F0F2F0F0 00000000   * .......2005041308155901020 0....*
40702260 47F0F022 01C3C5C5 00000118 00000370    47F0F001 58F0C31C 184E05EF 00000000   *.00..CEE.........00..0C..+......*
40702280 07F390EB D00C58E0 D04C4100 E1185500    C3144130 F03A4720 F01458F0 C28090F0   *.3..•..™•<......C...0...0..0B..0*

...

407025C0 42640000 00000000 6CF64BF3 86000000    1CCEA106 000003B0 00000000 00000000   *........%6.3f..................*
407025E0 FFFC0000 00000000 90000000 00400012    00000000 500001A8 FFFFFC90 38260000   *.............. ......&..y........*
40702600 400801A0 00000000 0006A281 F9F9F9F6    03002202 FFFFE988 00000000 FFFFFC38   * .........sa9996......Zh........*
40702620 00000000 00000000 58F0C210 58F0F158    07FF0000 00000000 E2F0F0F3 00E40101   *.........0B..01.........S003.U..*
40702640 00000000 00000000 00000000 00000000    00000000 00000000 00000000 00000000   *................................*
      LINES 40702660-407026E0  SAME AS ABOVE
```

Der Save Area Trace im SYSUDUMP (3)

Bei der Dump-Analyse mit SYSUDUMPs gibt es im Vergleich zu den CEEDUMPs folgende Schwierigkeiten:

- die Fehler- und Aufrufoffsets der verschiedenen Funktionsebenen werden nicht angezeigt,
- die C-Funktionsnamen werden nicht angezeigt.

Die DSA-Adressen, Entry Points, Rücksprung- und Parameteradressen sind kein Problem, abgesehen davon, dass im SYSUDUMP die Save Areas anhand der darin enthaltenen Entry Points beschriftet werden, im CEEDUMP jedoch anhand der DSA-Adresse. Das bezieht sich jedoch nur auf die Modulnamen; die C-Funktionsnamen werden ja ohnehin nicht korrekt angezeigt.

Die fehlenden Informationen lassen sich jedoch ermitteln bzw. berechnen:

- Fehlerstelle (bei INTERRUPT AT) minus Entry Point der aktiven Funktion ergibt den Fehleroffset (Typ c) analog CEEDUMP:

```
  Fehlerstelle im PSW                = 4070252A
- Entry Point der aktiven Funktion   = 40702260
-------------------------------------------------
= Fehler-Offset (Typ c)              = 000002CA
```

- die Aufruf-Offsets der übergeordneten Funktionen lassen sich aus der Distanz zwischen Entry Point und Rückkehradresse berechnen:

```
  Rückkehradresse                    = 407013C6
- Entry Point der übergeordn. Funkt. = 40701030
-------------------------------------------------
= Aufruf-Offset (Typ c)              = 00000396
```

- die Funktionsnamen stehen in den PPA1-Kontrollblöcken am Offset X'38'; im Wort an der Stelle Entry Point + 12 steht der Offset des PPA1-Kontrollblocks, bezogen auf den jeweiligen Entry Point:

```
  Entry Point der aktiven Funktion   = 40702260
+ Offset des PPA1-Kontrollblocks     = 00000370
-------------------------------------------------
= Adresse des PPA1-Kontrollblocks    = 407025D0
```

(frei für Notizen)