KERNFORSCHUNGSZENTRUM

KARLSRUHE

Februar 1974 KFK 1926

Institut für Datenverarbeitung in der Technik

Erstellung eines Programmsystems unter besonderer Berücksichtigung der Übertragbarkeitseigenschaften

H. Fichtel

GESELLSCHAFT
FÜR
KERNFORSCHUNG M.B.H.

KARLSRUHE

GESELLSCHAFT FÜR KERNFORSCHUNG M.B.H.
KARLSRUHE

KERNFORSCHUNGSZENTRUM KARLSRUHE

KFK 1926

Institut für Datenverarbeitung
in der Technik

Erstellung eines Programmsystems unter besonderer Berücksichtigung der Übertragbarkeitseigenschaften

H. Fichtel

Gesellschaft für Kernforschung mbH, Karlsruhe

Zusammenfassung

Ein wesentliches Problem bei der Erstellung von Software für kleine und mittlere Prozeßrechner ist das Problem der Übertragbarkeit. Für diese Rechnertypen existieren in der Regel keine standardisierten und kompatiblen höheren Programmiersprachen. In einer Fallstudie wird das Prinzip der Erstellung übertragbarer Software nach dem Konzept der abstrakten Maschine beschrieben und eine Aufwandsabschätzung vorgenommen.

Abstract

A general problem in the development of software for medium-sized and minicomputers is the problem of portability. Generally there are no standardized and compatible high-level programming languages for this class of computers. In a typical example, the development of portable software applying the concept of abstract machine modelling is described and evaluated.

Inhaltsverzeichnis

		Seite
1.	Einleitung	1
2.	Verwendung höherer Programmiersprachen	2
3.	Konzept der abstrakten Maschine	3
	3.1 Definition der abstrakten Maschine	3
	3.2 Lösung des Problems	3
	3.3 Implementierung der abstrakten Maschine	4
4.	Überlegungen zum Entwurf einer abstrakten Maschine	5
	4.1 Verhältnis abstrakte Maschine zum Problem	5
	4.2 Verhältnis abstrakte zur realen Maschine	6
	3.2 Verhältnis abstrakte Maschine zu den Hilfsmitteln der Implementierung	8
5.	Beschreibung einer Fallstudie	9
6.	Beschreibung der Implementierungssprache	1o
	6.1 Datenstrukturen	1o
	6.1.1 Register, Speicherzellen	1o
	6.1.2 Datentypen	11
	6.1.3 Zugriff auf Daten	11
	6.2 Operationen	13
	6.3 Kontrollstrukturen	14
	6.3.1 Sprünge, Marken	15
	6.3.2 Bedingte Anweisungen	16
	6.3.3 Schleifenanweisung	17
	6.3.4 Unterprogrammtechnik	17
	6.4 Ein-/Ausgabe	18
7.	Diskussion der Implementierungs- und Übertragungsaufwandes	18
	7.1 Implementierungsaufwand	18
	7.2 Übertragungsaufwand	19
8.	Literaturverzeichnis	21

1. Einleitung

Ein wesentliches Problem bei der Erstellung von Anwendersoftware ist das Problem der Portabilität oder Übertragbarkeit.
Portabilität wird gemessen an den Bemühungen, ein bereits ausgetestet existierendes und auf einem realen Rechner ablauffähiges Programm auf einen anderen realen Rechner zu übertragen. Wenn die Summe dieser Bemühungen insgesamt wesentlich geringer ist als der Arbeitsaufwand, das Programm in einer auf dem neuen Zielrechner verfügbaren Programmiersprache neu zu erstellen, ist es von hoher Portabilität oder gut übertragbar.

Ein Problem von ähnlicher Bedeutung ist das Problem der Adaptierbarkeit oder Änderungsfreundlichkeit. Adaptierbarkeit wird gemessen an den Bemühungen, ein bereits existierendes Programm, dessen Anforderungsprofile oder Problemspezifikationen sich von der Benutzerseite her geändert haben,so zu ändern, daß das Programm diesen neuen Anforderunen auf demselben Rechner gerecht wird.
Adaptierbare Programme sind somit wartungsfreundlich.

Portabilität und Adaptierbarkeit sind die zwei verschiedenen Seiten einer Münze (Programm):
Portabilität ist das Verhalten eines Programmsystems gegenüber wechselnden Maschinen, Adaptierbarkeit gegenüber wechselnden Benutzeranforderungen.

Die folgenden Ausführungen beschränken sich im wesentlichen auf das Konzept der Portabilität.
Die Motivation zur Erstellung gut übertragbarer Software im allgemeinen ist unmittelbar einleuchtend:

> Vermeiden unnötiger Doppelarbeit etwa bei der Umrüstung auf einen neuen Rechner.

Ein weiterer Grund speziell für Kleinrechner und kleine bis mittlere Prozeßrechner ist der folgende:

> für diese Klasse von Rechnern benötigte Software kann auf Großrechnern unter Ausnutzung dort vorhandener komfortabler Testhilfen erstellt und ausgetestet werden, bevor die Übertragung auf die eigentliche Zielmaschine erfolgt (half bootstrapping/6/,pulling /8/).

2. Verwendung höherer Programmiersprachen

Die traditionelle Methode zur Erzielung guter Übertragbarkeitseigenschaften von Programmsystemen ist die Verwendung von höheren Programmiersprachen wie FORTRAN und ALGOL. Diese Methode setzt dabei zwei Bedingungen voraus:

- die verwendete Programmiersprache ist dem Problem angemessen, d.h. ihre elementaren Datenstrukturen und Operationen gestatten eine passende Formulierung eines Algorithmus zur Problemlösung,
- die verwendete Programmiersprache ist für die in Frage kommenden Rechner tatsächlich verfügbar, und zwar in standardisierter Form.

Für die üblichen Fragestellungen in Wissenschaft und Technik (FORTRAN, ALGOL, PL/1) und im kommerziellen Bereich (COBOL, RPG) gibt es Programmiersprachen, die die erste Bedingung annähernd zufriedenstellend erfüllen. Insbesondere geht die Entwicklung moderner Sprachen wie ALGOL68 dahin, es dem Benutzer zu gestatten, sich eigene Datenstrukturen mit zugehörigen eigenen Operationen auf diese neuen Datenstrukturen in geeigneter Weise selbst zu definieren.

Die zweite Bedingung dagegen wird insbesondere für Klein- und Prozeßrechner in der Regel nicht erfüllt. Aus diesem Grunde sind Anwender von Prozeßrechnern gezwungen, große Teile ihrer Programm-

systeme in der Assemblersprache der verwendeten Maschine zu schreiben. Bei der Anschaffung neuer Rechner hat denn auch gerade dieser Anwenderkreis die entsprechenden Übertragungsschwierigkeiten, da Assemblersprachen keine der genannten Bedingungen für gute Portabilität erfüllen.

3. Konzept der abstrakten Maschine

Das Konzept der abstrakten Maschine als Methode zur Erstellung übertragbarer Software ist in der Literatur ausgiebig beschrieben worden /1 ... 5/. Es eignet sich insbesondere für den bereits angesprochenen Kreis kleinerer Rechner, bei denen die Verwendung höherer Programmiersprachen, wenn überhaupt vorhanden, mangels allgemein akzeptierter Standards nicht wesentlich zur Erhöhung der Übertragbarkeit beiträgt.

Vom Konzept her erfolgt die Erstellung eines neuen übertragbaren Programmsystems in drei Schritten.

3.1 Definition der abstrakten Maschine

Der erste Schritt ist die Definition geeigneter Datenstrukturen und entsprechender elementarer Operationen auf diese Datenstrukturen, mit denen das Problem in natürlicher Weise beschrieben werden kann. Zusätzlich wird eine im allgemeinen problemunabhängige Menge von Kontrollstrukturen definiert. Dieser Schritt wird von Poole und Waite /6/ als Definition einer 'abstrakten Maschine' und von Brown als Definition einer 'Beschreibungssprache' (descriptive language) bezeichnet. Genauere Überlegungen zum Entwurf dieser abstrakten Maschine werden im Abschnitt (4) gemacht.

3.2 Lösung des Problems

Der zweite Schritt ist das eigentliche Arbeitsgebiet des Anwendungsprogrammierers. Unter Verwendung der im ersten Schritt definierten

Datenstrukturen, Operationen und Kontrollstrukturen wird der problemlösende Algorithmus erstellt. Mit anderen Worten, das eigentliche Programmsystem wird in einer auf natürliche Weise auf das Problem zugeschnittenen Sprache der gerade definierten abstrakten Maschine erstellt.

Es ist wichtig zu betonen, daß bisher noch nichts über die Repräsentation der Datenstrukturen sowie die Art der Durchführung der Operationen auf dem Zielrechner gesagt wurde. Der Algorithmus ist vollständig unabhängig von der internen Darstellung der Datenstrukturen und der Weise, wie die Operationen zu ihrem Resultat gelangen.

3.3 Implementierung der abstrakten Maschine

Im dritten Schritt schließlich werden die für die Problemlösung als geeignet erachteten Operationen und Daten auf dem realen Rechner implementiert, auf dem das erstellte Programm ablaufen soll. Es wird also die abstrakte Maschine auf die reale Maschine abgebildet (Bild 1).

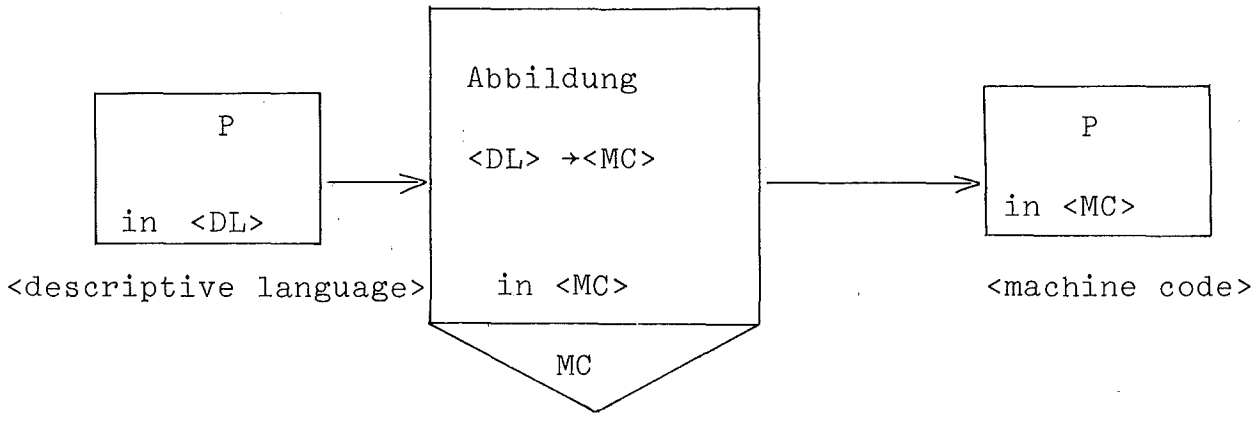

Bild 1: Übertragung des Programms P auf den Rechner mit der Sprache MC

Vom Konzept her gesehen existiert der Unterschied zwischen der Verwendung einer existierenden höheren Programmiersprache und der beschriebenen Methode, eine abstrakte Maschine zu definieren und auf den Zielrechner abzubilden, praktisch nicht.

Nur ist man im ersteren Fall auf die Vor- und Nachteile einer existierenden abstrakten Maschine, nämlich etwa der ALGOL- oder PL/1-Maschine angewiesen, die bereits beim Entwurf der Programmiersprache spezifiziert worden ist. Der erste Schritt ist dem Benutzer vom Entwickler der gewählten Programmiersprache und der dritte Schritt vom Implementierer abgenommen worden.

Letzteres ist als gewichtiges Argument für die Wahl einer existierenden Programmiersprache zu werten, sofern deren Implementierung nur hinreichend effizient und insbesondere auf Prozeßrechnern überhaupt vorhanden ist.

Ob die Vorwegnahme der abstrakten Maschinendefinition für die Wahl einer höheren Programmiersprache spricht, hängt vornehmlich vom Problem ab. Der Entwurf einer in weiten Problembereichen verwendbaren Sprache stellt natürlich einen Kompromiß zwischen den verschiedensten Anforderungen dar. So erlauben höhere Programmiersprachen in der Regel keinen Zugriff auf spezielle Maschineneigenschaften oder Peripherie.

4. Überlegungen zum Entwurf einer abstrakten Maschine

Der erste und folgenschwerste Schritt gemäß Abschnitt (3) ist die Spezifikation einer abstrakten Maschine. Deren Entwurf wird nicht nur den Aufwand bei der Problemlösung beeinflussen, sondern auch die Übertragung auf eine reale Maschine. Deshalb wird eine solche abstrakte Maschine nicht für einzelne Programme, sondern für einen weiteren Problemkreis entworfen, um das Verhältnis von Implementierungsaufwand zu produzierter Anwendersoftware in einer vertretbaren Relation zu halten.

Den Entwurf einer solchen Maschine beeinflussen insbesondere drei Überlegungen:

4.1 Das Verhältnis abstrakte Maschine zum Problemkreis

Der Entwurf der abstrakten Maschine spezifiziert die Weise, in der die speziellen anstehenden Probleme beschrieben und die problem-

lösenden Algorithmen kodiert werden.

Existierende Programmiersprachen sind in der Regel nur auf zwei Ebenen verfügbar, nämlich einem problem- oder einem maschinenabhängigen Niveau.
Problemabhängige Sprachen ermöglichen eine gut lesbare und strukturierte Formulierung des Problems und damit auch eine gute Dokumentation und Änderungsfreundlichkeit. Der Zugriff auf spezielle Maschineneigenschaften, wie etwa Unterbrechungsbehandlung, Bitmanipulation oder die Kontrolle spezieller Peripheriegeräte, ist dagegen entweder überhaupt nicht oder nur sehr ineffizient möglich.

Maschinenabhängige Sprachen wie Assemblersprachen erlauben ein hohes Maß an Kontrolle über die reale Maschine und die Benutzung aller Maschinenbefehle in freier Form. Das geht allerdings nur auf Kosten einer vergleichsweise undurchsichtigen Kodierung und zeitlich ausgedehnter Entwicklungs- und Testphasen.

Eine abstrakte Maschinensprache kann nun abhängig vom gestellten Problem beliebig zwischen diese beiden Niveaus gelegt werden, insbesondere was die Datenstrukturen und Operationen betrifft. Die Kontrollstrukturen sollten denen höherer Programmiersprachen nachempfunden werden, denn sie beeinflussen die Lesbarkeit und damit auch die Änderungsfreundlichkeit von Programmen sehr stark. Auf diese Weise können zumindest einige der Vorteile höherer Programmiersprachen mit der Flexibilität und dem hohen Maß an Kontrolle über die verfügbaren Maschinenoperationen von konventionellen Assemblersprachen verbunden werden.

4.2 Das Verhältnis der abstrakten zur realen Maschine

Im dritten Schritt gemäß Abschnitt (3) wird die abstrakte Maschinensprache auf die Sprache einer realen Maschine abgebildet. Der Implementierungsaufwand dieser Abbildung hängt natürlich ziemlich direkt davon ab, wie weit diese beiden Sprachen voneinander entfernt sind. Eine Spezifikation einer abstrakten Maschine, die auf die Struktur der realen Maschine wenig Rücksicht nimmt, bietet etwa die Vorteile

der höheren Programmiersprachen gegenüber Assemblersprachen:

- größere Maschinenunabhängigkeit
- höherer Programmierkomfort
- bessere Dokumentation und damit größere Änderungsfreundlichkeit
- kürzere Programmentwicklungs- und Testzeiten
- bessere Optimierungsmöglichkeiten des Objektcodes

Der Nachteil von abstrakten Maschinensprachen auf hohem Niveau ist ihre schwierige und zeitraubende Implementierung.

Dagegen ist die Implementierung niederer abstrakter Maschinensprachen einfach und schnell durchführbar. Als Folge davon haben Programme,die in einer solchen niederen Sprache geschrieben sind, eine höhere Übertragbarkeit.

In /5/ wird zur Implementierung der niederen Sprache FLUB auf eine beliebige Zielmaschine ein Zeitraum von etwa einer Mannwoche veranschlagt [+)]. In /7/ werden keine absoluten Zeiten angegeben, aber das Verhältnis des Implementierungsaufwandes der niederen Sprache LOWL zur höheren Sprache L wird auf mindestens 1:4 geschätzt.

Eine Methode, die Vorteile von abstrakten Maschinensprachen beiden Niveaus zu erzielen, ist die Einführung einer Hierarchie von abstrakten Maschinen /9/. Dabei wird das Programmsystem, das übertragen werden soll, in der problemorientierten Sprache der abstrakten Maschine M_n geschrieben. Diese Maschine M_n wird nun nicht direkt auf eine reale Maschine übertragen, sondern auf eine zweite abstrakte Maschine M_{n-1}, deren Niveau näher dem der realen Maschine liegt. Diese Hierarchie kann in beliebig vielen Stufen realisiert werden. Wenn jetzt alle abstrakten Maschinen M_i in Termen der jeweils niedrigeren Stufe M_{i-1} definiert sind, ist nur noch die letzte Stufe M_1 von der Übertragung betroffen. M_1 ist die Basismaschine der abstrakten Maschinenhierarchie und muß auf die reale Zielmaschine abgebildet werden.

+) FLUB ist die Sprache, in der der Makroprozessor STAGE2 geschrieben ist. Der angegebene Zeitraum gibt somit auch den Aufwand zur Übertragung dieses Makroprozessors auf eine andere Maschine an.

4.3 Das Verhältnis der abstrakten Maschine zu den Hilfsmitteln der Implementierung

Üblicherweise wird als Werkzeug zur Abbildung der Sprachkonstruktionen der abstrakten Maschine auf die der realen Maschine ein Makroprozessor benutzt, der selbst nach der gleichen Methode übertragbar ist /3 ...6/. Die Abbildungsvorschriften sind dann als Satz von Makros kodiert. Die Sprachkonstruktionen der abstrakten Maschine entsprechen den Makroköpfen, die den Aufruf der jeweiligen Makros steuern, und der erzeugte Objektcode für eine reale Maschine den Makrokörpern.

Besonders maschinenabhängige Operationen - wie etwa Ein-/Ausgabe, Codewandlung u.ä. - werden dabei in der Regel nicht als Inline-Objektcode, sondern als Unterprogrammsprünge in ein Laufzeitsystem realisiert. Dieses Laufzeitsystem ist natürlich maschinenabhängig und muß bei jeder Übertragung auf eine andere reale Maschine etwa in der entsprechenden Assemblersprache neu geschrieben werden.

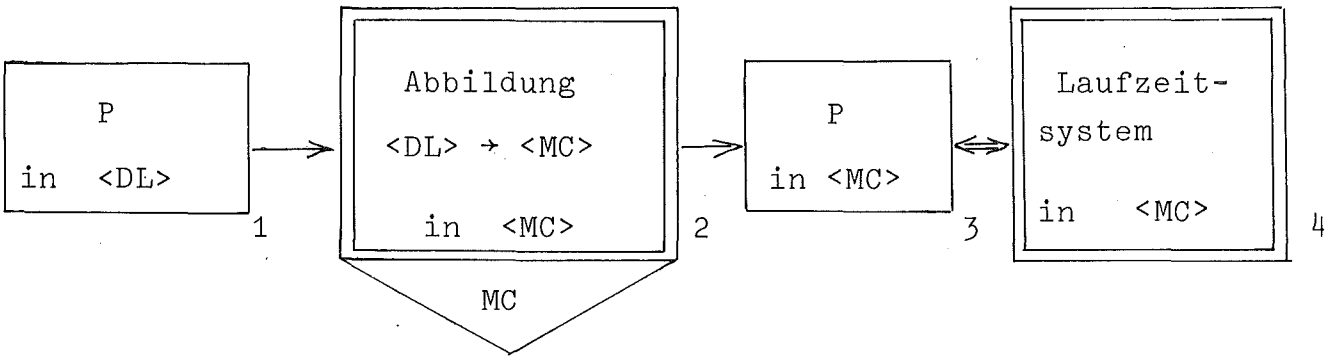

Bild 2: Übertragung des Programms P auf den Rechner mit der Sprache MC

Dieses Übertragungsschema zeigt Bild (2). Der Hauptvorteil dieser Technik besteht darin, daß das zur Übertragung anstehende Programmsystem konsequent in einen maschinenunabhängigen und einen maschinenabhängigen Teil getrennt ist. Den maschinenunabhängigen Teil repräsentiert das in einer abstrakten Maschinen-Sprache DL geschriebene Programm P (Kasten 1). Der maschinenabhängige Teil besteht aus zwei Komponenten, dem Makrosatz zur Abbildung (Kasten 2) und dem Laufzeit-

system (Kasten 4). Zum Ablauf auf einer realen Maschine werden das in den Objektcode übersetzte Programm P (Kasten 3) und das Laufzeitsystem zu einem ablauffähigen Programm zusammengebunden.

Ein Nachteil dieser Technik ist der bekannte hohe Aufwand an Speicherplatz und Rechenzeit, den Makroprozessoren erfordern. Der Lauf des Makroprozessors und damit die Erzeugung des Objektcodes ist daher in der Regel nur auf Großrechnern möglich.
Je nach Wahl des Makroprozessors können schon beim Entwurf der abstrakten Maschine spezielle Eigenschaften (etwa Formateinschränkungen) berücksichtigt werden, um den Implementierungsaufwand möglichst gering zu halten.

5. Beschreibung einer Fallstudie

Im Rahmen eines kurzfristigen Projekts war es das Ziel, ein Betriebs- und interaktives Programmiersystem zu erstellen, das insbesondere für Kleinrechner und kleinere Prozeßrechner gedacht war.

Das Betriebssystem besteht dabei aus einer einfachen Speicherverwaltung und Ein-/Ausgabeorganisation. Als Programmiersprache für das Programmiersystem wurde BASIC gewählt, weil diese Sprache eine einfache und speicherplatzsparende Implementierung erlaubt. Die Komponenten des Programmiersystems sind ein interaktiver Texteditor zur Programmerstellung und -wartung, ein interaktiver BASIC-Interpreter zum Programmtest, ein BASIC-Compiler zur Erzeugung von Objektcode für ausgetestete Programme sowie ein Binder, um getrennt übersetzte Programme laden und binden zu können.

Für die Erstellung dieses Programmsystems war die Forderung nach weitgehender Maschinenunabhängigkeit bzw. Übertragbarkeit zu berücksichtigen, weil das System auf einer Großrechenanlage mit seinen komfortablen Testmöglichkeiten erstellt und fertig ausgetestet werden sollte. Nach Ablauf dieser Erstellungsphase soll es mit möglichst wenigen Änderungen auf verschiedenen kleineren Rechnern ablauffähig sein.

Da für Maschinen dieser Größenordnung keine allgemein kompatible höhere Programmiersprache existiert, wurde zur Erzielung einer hohen Übertragbarkeit das Konzept der abstrakten Maschine gemäß Abschnitt (3) verwendet.

6. Beschreibung der Implementierungssprache

Für die Programmierung eines komplexen Programmsystems mit guten Übertragbarkeitseigenschaften gemäß dem Konzept der abstrakten Maschine wurde eine Implementierungssprache ASR[+)] definiert. Diese Sprache wird im folgenden in informaler Weise vorgestellt und beschrieben.

6.1 Datenstrukturen

Das in Abschnitt (5) beschriebene Programmsystem hat typische Aufgaben der Textverarbeitung zu erfüllen. Dementsprechend muß es die Implementierungssprache gestatten, relativ komplexe Datenstrukturen aufzubauen und darauf zugreifen zu können.

6.1.1 Register, Speicherzellen

Es werden 16 allgemeine arithmetische Register der Länge m Bit vorausgesetzt, die als Operand etwa in arithmetischen Ausdrücken angesprochen werden können. Sie werden durch Ri mit i=0...15 bezeichnet.

In der Implementierung für Registermaschinen wie IBM/36o oder Siemens 32o/33o können die 16 vorhandenen schnellen Register ausgenutzt werden. Bei Registermaschinen mit weniger als 16 allgemeinen Registern wie PDP 11/4o oder Akkumulatormaschinen wie Siemens 3o5 oder CD 33oo müssen die fehlenden Register als globale Variable im Kernspeicher simuliert werden.
Es sind keine Gleitkommaregister vorgesehen.

+) Assemblernahe Sprache für Registermaschinen

Die im Kernspeicher adressierbaren Einheiten sind ein Wort mit m Bit

z.B.	IBM/36o	m=32
	Siemens 32o/33o	m=16
	PDP 11	m=16
	CD 33oo	m=24

oder ein Zeichen mit n Bit

z.B.	IBM/36o	n=8
	Siemens 32o/33o	n=8
	PDP 11	n=8
	CD 33oo	n=6

6.1.2 Datentypen

An Datentypen existieren arithmetische Variable in oktaler, dezimaler und hexadezimaler Schreibweise sowie Textvariable. Jede Variable, die im Programm angesprochen wird, muß vom Programmierer vereinbart werden. Entsprechend den üblichen Konventionen von Assemblersprachen muß die Vereinbarung nicht lexikographisch vor der ersten Referenz stehen.In einer Vereinbarung wird der Name der Variablen definiert, der Platzbedarf angegeben und ggfs. eine Initialisierung vorgenommen. Die Platzreservierung für arithmetische Variable geschieht in Vielfachen eines Wortes, die für Textvariable in Vielfachen eines Zeichens.

```
z.B.  res arith size 2w;
      res platz size 8oc;
      def konst64 val d'64', o'1oo', h'4o';
      def text    val c'fehler in i/o';
```

6.1.3 Zugriff auf Daten

Der Zugriff auf im Kernspeicher stehende Daten geschieht entweder auf ein Wort oder auf ein Zeichen. Dafür stehen zwei Zuweisungsoperatoren zur Verfügung.

```
r2:=var;
```

bewirkt das Laden des Wortes mit der symbolischen Adresse var in das allgemeine Register 2, und

```
char::=c'*';
```

bewirkt das Speichern der Textkonstanten * in das Zeichen mit der symbolischen Adresse char.

Der Zugriff auf Felder wird ermöglicht mittels einer Adreßmodifikation durch Angabe entweder einer Konstanten oder eines Registers.
Die Vereinbarung eines Feldes erfolgt gemäß Abschnitt (6.1.2) durch die Angabe des Namens und des Gesamtplatzbedarfs. Das erste Element bekommt den Index 0 zugeordnet.

```
:
res feld size 10w;
:
r2:=7;
:
feld(r2):=0;
:
```

Das achte Wort im Vektor feld wird gleich null gesetzt.

Die Implementierung und der Zugriff auf eine Listen- oder Baumstruktur wird ermöglicht durch einen Adreßoperator @ und den Zugriff mittels indirekter Adressierung (Adreßsubstitution).

```
zeiger:= @feld;
```

bewirkt das Laden eines Zeigers auf das Feld feld und Abspeichern dieser Adresse in die Variable zeiger.

```
"zeiger"(r2):=1;
```

setzt das im vorletzten Beispiel zu null gesetzte achte Element des Feldes nunmehr zu 1.

6.2 Operationen

An Operationen stehen etwa alle in üblichen Assemblersprachen gebräuchlichen zur Verfügung.

Bereich	Operatoren
Arithmetik	+ - * /
Wortzuweisung	:=
Zeichenzuweisung	::=
Wortvergleich	.gt. .lt. .ge. .le. .eq. .ne.
Zeichenvergleich	.gtc. .ltc. .gec. .lec. .eqc. .nec.
Logik	.and. .or. .xor. .not.
Schieben	.sll. .srl. .sla. .sra.+)
Adresse	@
Wandlung	.db. .bd. .hb. .ob. .bo.

Tabelle 1: In ASR verfügbare Operatoren

Die in Tabelle (1) gelisteten Operatoren erklären sich im wesentlichen selbst.

Der linke Operand der binären Schiebeoperatoren spezifiziert die Zahl der Schiebeschritte, der rechte in einem Adreßausdruck den zu schiebenden Operanden.

```
"r10" := 2.sll. feld(r2);
```

Die Wandeloperatoren veranlassen die Wandlung von Zahlen von der internen Darstellung (b=binär) in die entsprechende Druckerdarstellung (d=dezimal, o=oktal, h=hexadezimal) für Zwecke der Ein-/Ausgabe. Der linke Operand spezifiziert dabei die Anzahl der auszugebenden Druckstellen und der rechte in einem Adreßausdruck den zu wandelnden Operanden.

+) .sll. shift left logical
.sra. shift right algebraical

```
text(8):=r2.bd. binzahl;
```

Es sind beliebig lange arithmetische Ausdrücke möglich. Eine Klammerung ist jedoch nicht erlaubt, und alle Operatoren haben dieselbe Priorität. Die Abarbeitung erfolgt von links nach rechts.

```
var:= -r2 * 5;
"zeiger" := "r1o"+2o * feld(3);
```

6.3 Kontrollstrukturen

In ASR sind einige in höheren Programmiersprachen übliche Kontrollstrukturen definiert worden. Diese Konstruktionen erhöhen die Lesbarkeit eines Programms erheblich und tragen somit auch zur Änderungs- und Wartungsfreundlichkeit bei.

Das folgende Programmbeispiel soll einen informellen Eindruck davon vermitteln.

Die Routine suchen sucht und findet ein Element in einer verketteten Liste, das eine bestimmte Information (den Suchbegriff) enthält, falls dieses Element existiert. Im Register 0 erwartet die Routine den Suchbegriff, in Register 1 einen Zeiger auf den Anfang der Liste. Bei Erfolg wird im Register der Zeiger auf das gesuchte Element zurückgeliefert, bei Mißerfolg eine negative Zahl.

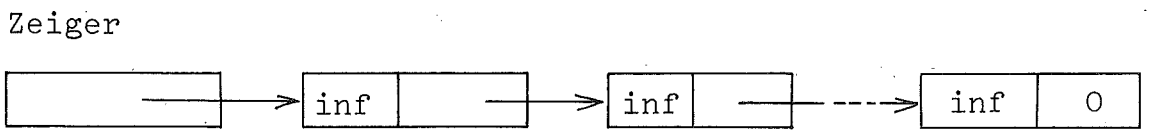

Bild 3: Verkettete Liste

```
suchen: proc;
schleife: if "r1" .eq. ro then;
          return;
          else;
               if "r1"(1) .eq. o then;
               r1:= -1;
               return;
               fi;
               r1:= "r1"(1);
          fi;
          goto schleife;
```

Ein Aufruf der Routine würde etwa erfolgen durch

```
ro:= suchbegriff;
r1:= zeiger;
call suchen;
if r1 .eq. -1 then;
   goto fehler;
fi;
:
fehler:
:
```

6.3.1 Sprünge, Marken

Marken werden auf die übliche Weise definiert, indem sie, durch einen Doppelpunkt getrennt, vor die Anweisung geschrieben werden.

```
marke: ....;
```

Auf diese Marken kann dann durch unbedingte Sprünge verzweigt werden.

```
goto marke;
```

6.3.2 Bedingte Anweisungen

Die bedingte Anweisung kann ohne Alternative

```
if a-1 .ne. b(r2) then;
        :
fi;
```

oder auch mit Alternative

```
if c'*' .eqc. b(r2) then;
        :
else;
        :
fi;
```

Das Problem der Zweideutigkeit des 'dangling else' von ALGOL60 wird durch Übernahme des klammernden fi von ALGOL68 vermieden.

Eine Eigenheit der Notation von ASR kommt hier deutlich zum Ausdruck, nämlich daß die drei syntaktischen Einheiten if ... then, else und fi in einer Zeile für sich allein stehen müssen. Dies ist eine Einschränkung zur Erleichterung der Implementierung gemäß Abschnitt (4.3). Die Ursache liegt im verwendeten Hilfsmittel, dem Makroprozessor STAGE2 /5/, begründet. Die Eingabe zum STAGE2 ist zeilenweise organisiert. Nach jedem Einlesen einer Zeile wird ein Makro gesucht, das von dieser Zeile aufgerufen wird. Zur Vereinfachung nun wird die bedingte Anweisung durch drei Makros implementiert, die miteinander über einen Kellermechanismus kommunizieren.

Aus dem gleichen Grund wird jeweils nur eine Anweisung pro Zeile geschrieben und durch ein Semikolon abgeschlossen. Abgesehen von der Schleifenanweisung, bei der das gleiche Problem auftritt, ist die Notation der Sprache ansonsten formatfrei. Insbesondere haben Leerzeichen (blanks) keine Signifikanz.

6.3.3 Schleifenanweisung

Die Schleifenanweisung erfordert die Angabe von drei Registern, die mit dem Anfangswert der Schleifenvariablen, dem Inkrement und dem Endwert als beliebige arithmetische Ausdrücke geladen werden. Der Test wird vor dem ersten Schleifendurchlauf ausgeführt, d.h. die Schleife kann auch keinmal durchlaufen werden. Das Schleifenende wird durch das Wortsymbol od gekennzeichnet.

```
do  r2:=6 by r5:=a+2 to  r3:="zeiger"+1;
                :
od;
```

6.3.4 Unterprogrammtechnik

Durch die Einführung einer Unterprogrammtechnik wird eine einheitliche Schnittstelle für alle Unterprogramme definiert. Die Parameterübergabe erfolgt dabei wie in Assemblersprachen implizit über die Register. Sowohl das rufende als auch das gerufene Programm wissen, in welchen Registern sich welche Ein- bzw. Ausgabeparameter befinden und welche Bedeutung sie haben. Außer diesen zur Parameterübergabe verwendeten Registern werden keine weiteren Register verändert. Die Inhalte der vom gerufenen Unterprogramm verwendeten Arbeitsregister werden vor dem Überschreiben in einen Laufzeitkeller gerettet und vor dem Rücksprung wieder geladen. Insbesondere wird die Rückkehradresse gerettet, so daß ein rekursiver Unterprogrammaufruf möglich ist.

```
up: proc (r7,r9,r4);
        :
    return (r7,r9,r4);
```

Es wird eine Routine up definiert, die vor Ausführung ihrer ersten Anweisung ihre Rückkehradresse und die in der Liste spezifizierten Register rettet und vor dem Rücksprung wieder lädt.

Der Aufruf erfolgt durch

```
call up;
```

6.4 Ein-/Ausgabe

Die Eingabe und die Ausgabe der ASR-Maschine ist zeilenweise organisiert und erfolgt über logische Einheiten, die wie in FORTRAN durch ganze Zahlen spezifiziert werden. Die Angabe, welche physikalische Einheit angesprochen werden soll, erfolgt außerhalb des Programms, etwa durch die Steuersprache.
Zusätzlich zur Angabe der logischen Einheit wird in einem Adreßausdruck die Kernspeicheradresse angegeben, für die die Ein-/Ausgabe erfolgen soll, sowie die Anzahl der Zeichen.

```
put unit=1,adr=outbuf,lng=80;
get unit=7,adr=inbuf+6,lng=r2;
```

Die Ein-/Ausgabe selbst erfolgt binär, so daß etwaige erforderliche Wandlungen vom Programm selbst mit Hilfe der Wandeloperatoren vorgenommen werden müssen.

7. Diskussion des Implementierungs- und Übertragungsaufwandes

Die Erstellung übertragbarer Software nach dem Konzept der abstrakten Maschine erfordert einen Mehraufwand durch die erforderliche Abbildung der abstrakten auf die reale Maschine.

7.1 Implementierungsaufwand

Z.Zt. der Abfassung dieses Aufsatzes wird das in Abschnitt (5) beschriebene Programmsystem in der in Abschnitt (6) vorgestellten Implementierungssprache erstellt. Der Test soll auf der Großrechenanlage IBM 370/165 erfolgen, so daß die abstrakte Maschine zunächst auf diesen realen Rechner übertragen wurde. Als Hilfsmittel diente dazu der Makroprozessor STAGE2 /5/, der selbst übertragbar ist und auf der realen Maschine zur Verfügung steht.

Zur Übertragung des Programmsystems auf die IBM 370/165 wurden gemäß Bild (2) in Abschnitt (4.3) zwei maschinenabhängige Systeme erstellt.

Zum ersten wurde ein Satz von Makros erstellt, der die Implementierungssprache ASR auf die Zielsprache 360/Assembler abbildet.

Diese Abbildung geht in zwei Stufen vor sich. In der ersten Stufe wird eine ASR-Zeile in eine interne Zwischenform gewandelt und an die zweite Stufe übergeben, die daraus den gewünschten Objektcode erzeugt. Das entspricht einer zweistufigen abstrakten Maschinenhierarchie gemäß Abschnitt (4.2) und setzt den Übertragungsaufwand auf eine andere Zielmaschine erheblich herab. Bei einem Austausch der Zielmaschine muß nicht mehr der gesamte Makrosatz ausgetauscht werden, sondern nur noch der zweite Teil, der den eigentlichen Objektcode erzeugt.

Zum zweiten wurde ein Laufzeitsystem im 360/Assembler erstellt, das im wesentlichen die Wandeloperatoren von ASR implementiert und die eigentliche Ein-/Ausgabe durchführt.

Für die Erstellung des Makrosatzes, d.h. die Implementierung von ASR auf dem System /360 waren einschließlich einer Laufzeit- und Speicheroptimierung des Makroprozessors STAGE2 ca. 6 Mannmonate erforderlich. Zur Erstellung des Laufzeitsystems wurden 2 Mannwochen benötigt. Somit beträgt der Mehraufwand bei der Erstellung des Programmsystems durch die Berücksichtigung der Übertragungsaspekte ca. 6 1/2 Mannmonate.

Insbesondere ist noch zu betonen, daß der Test der Abbildung der abstrakten auf die reale Maschine nicht die Fertigstellung des eigentlich zu übertragenden Programmsystems erfordert, sondern durch eigens dafür hergestellte Testprogramme geschieht.

7.2 Übertragungsaufwand

Um den tatsächlich entstehenden Übertragungsaufwand auf eine andere reale Maschine abzuschätzen, wurde die abstrakte ASR-Maschine auf den Rechner CD 3300 abgebildet. Dieser Rechner wurde deshalb ausgewählt, weil seine Assemblersprache COMPASS und das System MSOS von früheren Arbeiten her bekannt waren, so daß in dieser Hinsicht keine Einarbeitungsphase erforderlich war.

Es wird noch einmal betont, daß für die neue Abbildung nicht mehr der gesamte Makrosatz neu geschrieben werden mußte, sondern nur noch der zweite Teil, der jetzt die interne Zwischenform nach COMPASS abzubilden hatte. Für diese Aufgabe ergab sich ein Aufwand von einer Mannwoche. Dabei sind etwaige Testzeiten nicht berücksichtigt, da der Test nicht durchgeführt werden konnte, weil z.Zt. keine CD 33oo zur Verfügung steht. Ebenso müßte noch ein neues Laufzeitsystem in COMPASS erstellt werden. Der gesamte zur vollständigen Übertragung des Programmsystems entstehende Aufwand kann insgesamt auf einen Mannmonat geschätzt werden.

Mit diesem Ergebnis dürfte ein Nachweis für die praktische Anwendbarkeit des Konzepts der abstrakten Maschine erbracht worden sein. Bei der Erstellung eines Programmsystems ist ein gewisser Mehraufwand zu leisten, der im geschilderten Fall bei 6 Mannmonaten lag. Für ein kleines Programm wird sich dieser Mehraufwand sicher nicht auszahlen. Komplexe Programmsysteme können jedoch mit dieser Methode ohne großen Zusatzaufwand auf andere Zielmaschinen übertragen werden.

Literatur

/1/ Wilkes, M.V.
An Experiment with a Self-compiling Compiler for a Simple List-processing Language
in: Annual Review of Automatic Programming Vol. 4, Pergamon Press, Oxford 1964

/2/ Brown, P.J.
The ML/1 Macro Processor
Comm. ACM Vol. 1o, No. 1o (Oct. 1967), pp. 618-623

/3/ Griswold, R.E.
A Guide to the Macro Implementation of SNOBOL4
Bell Telephone Labs., Murray Hill, N.J. 1969

/4/ Brown, P.J.
Using a Macro Processor to Aid Software Implementation
Comp.J. Vol. 12, No. 4 (Nov. 1969), pp. 327-331

/5/ Waite, W.M
The Mobile Programming System: STAGE2
Comm. ACM Vol. 13, No. 7 (July 197o) pp. 415-421

/6/ Newey, M.C.
Poole, P.C.
Waite, W.M.
Abstract Machine Modelling to Produce Portable Software -
A Review and Evaluation
Software - Practice and Experience Vol. 2 (1972), pp. 1o7-136

/7/ Brown, P.J. Levels of Language for Portable Software
Comm. ACM, Vol. 15, No. 12 (Dec. 1972)
pp. 1o59-1o62

/8/ Richards, M. The Portability of the BCPL Compiler
Software-Practice and Experience
Vol. 1 (1971), pp. 135-146

/9/ Poole, P.C. Hierarchical Abstract Machines
Proc. of the Culham Symposium on Software
Engineering, 1971

Anhang:

		Seite
A1.	Ein Beispielprogramm in ASR	24
A2.	Ergebnisse des Laufs	27
A3.	Erzeugter Code für IBM/36o	29
A4.	Erzeugter Code für CDC 33oo	4o

```
**;                                                                  00000010
**;                                                                  00000020
**  EIN BEISPIELPROGRAMM IN ASR;                                     00000030
**;                                                                  00000040
**  PSEUDO-ZUFALLSZAHLENGENERATOR;                                   00000050
**;                                                                  00000060
**;                                                                  00000070
PRG ZUFALL;                                                         00000080
**;                                                                  00000090
MODUL PROLOG;                                                       00000100
GOTO HAUPT .IN. HAUPT;                                              00000110
ENDMODUL;                                                           00000120
**;                                                                  00000130
MODUL ROUTINEN;                                                     00000140
ENTRY RANDOM,AUSWERT,AUSW3,AUSDRUK;                                 00000150
**;                                                                  00000160
**;                                                                  00000170
**  ZUFALLSZAHLENGENERATOR NACH DER;                                 00000180
**  MULTIPLIKATIVEN KONGRUENZMETHODE;                                00000190
**;                                                                  00000200
**  MULTIPLIKAND 2**16+5 (65541);                                    00000210
**  DIVISOR 2**32;                                                   00000220
**;                                                                  00000230
**  ZUFALLSZAHL ZWISCHEN 0 UND 9 WIRD;                               00000240
**  IN REG. 2 UEBERGEBEN;                                            00000250
**;                                                                  00000260
RANDOM:   PROC;                                                     00000270
          R2:=RAND1;                                                00000280
          R2:=R2*65541;                                             00000290
          RAND1:=R2;                                                00000300
**;                                                                 00000310
          R2:=R2*10;                                                00000320
          R0:=R0+5;                                                 00000330
          R2:=R0;                                                   00000340
          RETURN;                                                   00000350
**;                                                                 00000360
DEF RAND1 VAL D'8193';                                              00000370
**;                                                                 00000380
**;                                                                 00000390
**  ROUTINE ZUR AUSWERTUNG DES;                                      00000400
**  ZUFALLSZAHLENGENERATORS;                                         00000410
**;                                                                 00000420
**  ZUFALLSZAHL IST IN REG. 2;                                       00000430
**;                                                                 00000440
AUSWERT: PROC;                                                      00000450
          IF R2 .GE. 10 THEN;                                       00000460
             PUT UNIT=1,ADR=AUSW1,LNG=20;                           00000470
             GOTO AUSW2;                                            00000480
          ELSE;                                                     00000490
             AUSW3(R2):=AUSW3(R2)+1;                                00000500
          FI;                                                       00000510
AUSW2:    RETURN;                                                   00000520
**;                                                                 00000530
DEF AUSW1 VAL C' FEHLER IN RANDOM   ';                              00000540
DEF AUSW3 VAL 10D'0';                                               00000550
**;                                                                 00000560
**;                                                                 00000570
**  ROUTINE ZUM AUSDRUCKEN DER STATISTIK;                            00000580
**;                                                                 00000590
```

```
**  TABELLE STEHT BEI ADRESSE AUSW3;                                    00000600
**;                                                                     00000610
AUSDRUK: PROC(R2,R3,R4,R5);                                            00000620
**;                                                                     00000630
**  DRUCKE TITELZEILE;                                                 00000640
**;                                                                     00000650
        PUT UNIT=1,ADR=AUSD3,LNG=4;                                    00000660
        PUT UNIT=1,ADR=AUSD1,LNG=32;                                   00000670
        PUT UNIT=1,ADR=AUSD2,LNG=32;                                   00000680
        PUT UNIT=1,ADR=AUSD3,LNG=4;                                    00000690
**;                                                                     00000700
**  DRUCKSCHLEIFE;                                                     00000710
**;                                                                     00000720
        R5:=@AUSW3 .IN. ROUTINEN;                                      00000730
        DO R2:=0 BY R3:=1 TO R4:=9;                                    00000740
          AUSD4:="R5"(R2);                                             00000750
          AUSD5:=8 .BD. AUSD4;                                         00000760
          AUSD5:=C'    ';                                              00000770
          AUSD8(3)::=AUSD7(R2);                                        00000780
          PUT UNIT=1,ADR=AUSD8,LNG=16;                                 00000790
        OD;                                                            00000800
        RETURN(R2,R3,R4,R5);                                           00000810
**;                                                                     00000820
DEF AUSD1 VAL C' STATISTIK DES ZUFALLSGE';                             00000830
DEF       VAL C'NERATORS';                                             00000840
DEF AUSD2 VAL 32C'*';                                                  00000850
DEF AUSD3 VAL 4C' ';                                                   00000860
RES AUSD4 SIZE 1W;                                                     00000870
DEF AUSD8 VAL 8C' ';                                                   00000880
RES AUSD5 SIZE 2W;                                                     00000890
DEF AUSD7 VAL C'0123456789';                                           00000900
**;                                                                     00000910
**;                                                                     00000920
ENDMODUL;                                                              00000930
**;                                                                     00000940
**;                                                                     00000950
**  HAUPTPROGRAMM;                                                     00000960
**;                                                                     00000970
MODUL HAUPT;                                                           00000980
**;                                                                     00000990
**  SIND DIE ZAHLEN ZU DRUCKEN?;                                       00001000
**;                                                                     00001010
        PUT UNIT=1,ADR=HAUPT1,LNG=28;                                  00001020
        GET UNIT=4,ADR=HAUPT2,LNG=4;                                   00001030
**;                                                                     00001040
**  LIES ANZAHL DER ZAHLEN EIN;                                        00001050
**;                                                                     00001060
        PUT UNIT=1,ADR=HAUPT4,LNG=36;                                  00001070
        GET UNIT=4,ADR=BUFFER(1),LNG=4;                                00001080
        BUFFER:=C'0000';                                               00001090
        BUFFER:=8 .DB. BUFFER;                                         00001100
**;                                                                     00001110
**  AUSFUEHRUNGSSCHLEIFE;                                              00001120
**;                                                                     00001130
        DO R3:=1 BY R4:=1 TO R5:=BUFFER;                               00001140
          CALL RANDOM .IN. ROUTINEN;                                   00001150
          BUFFER:=R2;                                                  00001160
          R9:=8;                                                       00001170
          BUFFER:=R9 .BD. BUFFER;                                      00001180
```

```
            DO R6:=0 BY R7:=1 TO R8:=6;                                    00001190
              BUFFER(R6)::=C' ';                                           00001200
            OD;                                                            00001210
            IF HAUPT2 .EQ. HAUPT5 THEN;                                    00001220
              GOTO HAUPT3;                                                 00001230
            FI;                                                            00001240
            PUT UNIT=1,ADR=BUFFER,LNG=8;                                   00001250
HAUPT3:     CALL AUSWERT .IN. ROUTINEN;                                    00001260
          OD;                                                              00001270
**;                                                                        00001280
**  DRUCKE DIE STATISTIK;                                                  00001290
**;                                                                        00001300
          CALL AUSDRUK .IN. ROUTINEN;                                      00001310
          GOTO END;                                                        00001320
**;                                                                        00001330
RES BUFFER SIZE 2W;                                                        00001340
DEF HAUPT1 VAL C' SIND DIE ZAHLEN ZU DRUCKEN?';                            00001350
RES HAUPT2 SIZE 4C;                                                        00001360
DEF HAUPT4 VAL C' WIEVIELE ZAHLEN';                                        00001370
DEF        VAL C' SIND ZU GENERIEREN?';                                    00001380
DEF HAUPT5 VAL C'NEIN';                                                    00001390
**;                                                                        00001400
**;                                                                        00001410
END:      ENDPRG;                                                          00001420
```

```
SIND DIE ZAHLEN ZU DRUCKEN?
WIEVIELE ZAHLEN SIND ZU GENERIEREN?
       6
       7
       8
       0
       1
       5
       1
       7
       9
       4
       1
       4
       4
       5
       0
       9
       2
       8
       1
       1
       7
       6
       3
       9
       9
       0
       4
       7
       9
       1
       4
       4
       1
       2
       8
       9
       7
       9
       9
       0

STATISTIK DES ZUFALLSGENERATORS
**********************************

 0          0004
 1          0007
 2          0002
 3          0001
 4          0006
 5          0002
 6          0002
 7          0005
 8          0003
 9          0008
```

```
SIND DIE ZAHLEN ZU DRUCKEN?
WIEVIELE ZAHLEN SIND ZU GENERIEREN?

STATISTIK DES ZUFALLSGENERATORS
******************************

 0          0460
 1          0499
 2          0510
 3          0485
 4          0526
 5          0497
 6          0509
 7          0517
 8          0481
 9          0516



SIND DIE ZAHLEN ZU DRUCKEN?
WIEVIEL ZAHLEN SIND ZU GENERIEREN?

STATISTIK DES ZUFALLSGENERATORS
******************************

 0          0955
 1          1007
 2          1012
 3          0979
 4          1027
 5          0973
 6          1051
 7          1034
 8          0955
 9          1006
```

EXTERNAL SYMBOL DICTIONARY

16.37 12/12/73

SYMBOL	TYPE	ID	ADDR	LENGTH	LD ID
ZUFALL	SD	01	000000	000038	
PROLOG	LD		000024		01
SAVEPT@	ER	02			
HAUPT	ER	03			
ROUTINEN	SD	04	000038	000254	
RANDOM	LD		000038		04
AUSWERT	LD		00006C		04
AUSW3	LD		0000D0		04
AUSDRUK	LD		0000F8		04
OUTPUT@	ER	05			
AUSW3	ER	06			
BD@	ER	07			
HAUPT	SD	08	000290	0001ED	
DB@	ER	09			
RANDOM	ER	0A			
ROUTINEN	ER	0B			
AUSWERT	ER	0C			
AUSDRUK	ER	0D			
SAVEPT@	SD	0E	000480	000194	

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                    F01MAY72  12/12/73

                                       1 **K
                                       2 **K
                                       3 **K  EIN BEISPIELPROGRAMM IN ASR
                                       4 **K
                                       5 **K  PSEUDO-ZUFALLSZAHLENGENERATOR
                                       6 **K
                                       7 **K
000000                                 8 ZUFALL   CSECT
000000                                 9          USING *,15
000000 47FF 000C              0000C   10          B      12(15)
000004 07E9E4C6C1D3D340               11          DC     X'07',CL7'ZUFALL'
00000C 90EC D00C              0000C   12          STM    14,12,12(13)
000010 181D                           13          LR     1,13
000012 58D0 F030              00030   14          L      13,=V(SAVEPT@)
000016 58DD 0000              00000   15          L      13,0(13)
00001A 501D 0000              00000   16          ST     1,0(13)
00001E 41DD 0004              00004   17          LA     13,4(13)
                                      18 **K
                                      19          ENTRY PROLOG
000022 05F0                           20          BALR   15,0
000024                                21          USING *,15
000024                                22 PROLOG    EQU    *
000024 58E0 F010              00034   23          L      14,=V(HAUPT)
000028 58F0 F010              00034   24          L      15,=V(HAUPT)
00002C 07FE                           25          BR     14
000030                                26          LTORG
000030 00000000                       27                 =V(SAVEPT@)
000034 00000000                       28                 =V(HAUPT)
                                      29 **K
000038                                30 ROUTINEN CSECT
000038                                31          USING *,15
                                      32          ENTRY RANDOM
                                      33          ENTRY AUSWERT
                                      34          ENTRY AUSW3
                                      35          ENTRY AUSDRUK
                                      36 **K
                                      37 **K
                                      38 **K  ZUFALLSZAHLENGENERATOR NACH DER
                                      39 **K  MULTIPLIKATIVEN KONGRUENZMETHODE
                                      40 **K
                                      41 **K  MULTIPLIKAND 2**16+5 (65541)
                                      42 **K  DIVISOR 2**32
                                      43 **K
                                      44 **K  ZUFALLSZAHL ZWISCHEN 0 UND 9 WIRD
                                      45 **K  IN REG. 2 UEBERGEBEN
                                      46 **K
000038                                47 RANDOM    EQU    *
000038 50ED 0000              00000   48          ST     14,0(13)
00003C 41DD 0004              00004   49          LA     13,4(13)
000040 5820 F030              00068   50          L      2,RAND1
000044 1812                           51          LR     1,2
000046 5C00 F218              00250   52          M      0,=F'65541'
00004A 1821                           53          LR     2,1
00004C 5020 F030              00068   54          ST     2,RAND1
                                      55 **K
000050 1812                           56          LR     1,2
000052 5C00 F21C              00254   57          M      0,=F'10'
000056 1821                           58          LR     2,1
000058 5A00 F220              00258   59          A      0,=F'5'
00005C 1820                           60          LR     2,0
```

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                    F01MAY72  12/12/73

00005E 58D0 F224            0025C    61           S     13,=F'4'
000062 58ED 0000            00000    62           L     14,0(13)
000066 07FE                          63           BR    14
                                     64 **K
000068                               65 RAND1     DS    0F
000068 00002001                      66           DC    F'8193'
                                     67 **K
                                     68 **K
                                     69 **K  ROUTINE ZUR AUSWERTUNG DES
                                     70 **K  ZUFALLSZAHLENGENERATORS
                                     71 **K
                                     72 **K  ZUFALLSZAHL IST IN REG. 2
                                     73 **K
00006C                               74 AUSWERT  EQU   *
00006C 50ED 0000            00000    75           ST    14,0(13)
000070 41DD 0004            00004    76           LA    13,4(13)
000074 1802                          77           LR    0,2
000076 5810 F21C            00254    78           L     1,=F'10'
00007A 1901                          79           CR    0,1
00007C 4740 F066            0009E    80           BL    L@0000
000080 5810 F228            00260    81           L     1,=F'1'
000084 4100 F084            000BC    82           LA    0,AUSW1
000088 58B0 F22C            00264    83           L     11,=F'20'
00008C 18CF                          84           LR    12,15
00008E 58F0 F230            00268    85           L     15,=V(OUTPUT@)
000092 05EF                          86           BALR  14,15
000094 18FC                          87           LR    15,12
000096 47F0 F078            000B0    88           B     AUSW2
00009A 47F0 F078            000B0    89           B     L@0001
00009E                               90 L@0000    EQU   *
00009E 18C2                          91           LR    12,2
0000A0 88C0 0002            00002    92           SLA   12,2
0000A4 580C F098            000D0    93           L     0,AUSW3(12)
0000A8 5A00 F228            00260    94           A     0,=F'1'
0000AC 500C F098            000D0    95           ST    0,AUSW3(12)
0000B0                               96 L@0001    EQU   *
0000B0                               97 AUSW2     EQU   *
0000B0 58D0 F224            0025C    98           S     13,=F'4'
0000B4 58ED 0000            00000    99           L     14,0(13)
0000B8 07FE                         100           BR    14
                                    101 **K
0000BC                              102 AUSW1     DS    0F
0000BC 40C6C5C8D3C5D940             103           DC    CL20' FEHLER IN RANDOM
0000D0                              104 AUSW3     DS    0F
0000D0 0000000000000000             105           DC    10F'0'
                                    106 **K
                                    107 **K
                                    108 **K  ROUTINE ZUM AUSDRUCKEN DER STATISTIK
                                    109 **K
                                    110 **K  TABELLE STEHT BEI ADRESSE AUSW3
                                    111 **K
0000F8                              112 AUSDRUK   EQU   *
0000F8 50ED 0000            00000   113           ST    14,0(13)
0000FC 502D 0004            00004   114           ST    2,4(13)
000100 503D 0008            00008   115           ST    3,8(13)
000104 504D 000C            0000C   116           ST    4,12(13)
000108 505D 0010            00010   117           ST    5,16(13)
00010C 41DD 0014            00014   118           LA    13,20(13)
                                    119 **K
                                    120 **K  DRUCKE TITELZEILE
```

```
                                    121 **K
000110 5810 F228                00260 122           L     1,=F'1'
000114 4100 F1F0                00228 123           LA    0,AUSD3
000118 58B0 F224                0025C 124           L     11,=F'4'
00011C 18CF                           125           LR    12,15
00011E 58F0 F230                00268 126           L     15,=V(OUTPUT@)
000122 05EF                           127           BALR  14,15
000124 18FC                           128           LR    15,12
000126 5810 F228                00260 129           L     1,=F'1'
00012A 4100 F1B0                001E8 130           LA    0,AUSD1
00012E 58B0 F234                0026C 131           L     11,=F'32'
000132 18CF                           132           LR    12,15
000134 58F0 F230                00268 133           L     15,=V(OUTPUT@)
000138 05EF                           134           BALR  14,15
00013A 18FC                           135           LR    15,12
00013C 5810 F228                00260 136           L     1,=F'1'
000140 4100 F1D0                00208 137           LA    0,AUSD2
000144 58B0 F234                0026C 138           L     11,=F'32'
000148 18CF                           139           LR    12,15
00014A 58F0 F230                00268 140           L     15,=V(OUTPUT@)
00014E 05EF                           141           BALR  14,15
000150 18FC                           142           LR    15,12
000152 5810 F228                00260 143           L     1,=F'1'
000156 4100 F1F0                00228 144           LA    0,AUSD3
00015A 58B0 F224                0025C 145           L     11,=F'4'
00015E 18CF                           146           LR    12,15
000160 58F0 F230                00268 147           L     15,=V(OUTPUT@)
000164 18FC                           148           LR    15,12
                                    149 **K
                                    150 **K  DRUCKSCHLEIFE
                                    151 **K
000166 5850 F238                00270 152           L     5,=V(AUSW3)
00016A 5820 F23C                00274 153           L     2,=F'0'
00016E 5830 F228                00260 154           L     3,=F'1'
000172 5840 F240                00278 155           L     4,=F'9'
000176                                156 L@0002      EQU   *
000176 1924                           157           CR    2,4
000178 4720 F194                001CC 158           BH    L@0003
00017C 18C2                           159           LR    12,2
00017E 8BC0 0002                00002 160           SLA   12,2
000182 5805 C000                00000 161           L     0,0(5,12)
000186 5000 F1F4                0022C 162           ST    0,AUSD4
00018A 5810 F1F4                0022C 163           L     1,AUSD4
00018E 5800 F244                0027C 164           L     0,=F'8'
000192 41B0 F200                00238 165           LA    11,AUSD5
000196 18CF                           166           LR    12,15
000198 58F0 F248                00280 167           L     15,=V(BD@)
00019C 05EF                           168           BALR  14,15
00019E 18FC                           169           LR    15,12
0001A0 5800 F24C                00284 170           L     0,=CL4'
0001A4 5000 F200                00238 171           ST    0,AUSD5
0001A8 4302 F208                00240 172           IC    0,AUSD7(2)
0001AC 4200 F1FB                00233 173           STC   0,AUSD8+3
0001B0 5810 F228                00260 174           L     1,=F'1'
0001B4 4100 F1F8                00230 175           LA    0,AUSD8
0001B8 58B0 F250                00288 176           L     11,=F'16'
0001BC 18CF                           177           LR    12,15
0001BE 58F0 F230                00268 178           L     15,=V(OUTPUT@)
0001C2 05EF                           179           BALR  14,15
0001C4 18FC                           180           LR    15,12
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                  F01MAY72  12/12/73

0001C6 1A23                         181           AR     2,3
0001C8 47F0 F13E             00176  182           B      L@0002
0001CC                              183 L@0003    EQU    *
0001CC 5800 F22C             00264  184           S      13,=F'20'
0001D0 582D 0004             00004  185           L      2,4(13)
0001D4 583D 0008             00008  186           L      3,8(13)
0001D8 584D 000C             0000C  187           L      4,12(13)
0001DC 585D 0010             00010  188           L      5,16(13)
0001E0 58ED 0000             00000  189           L      14,0(13)
0001E4 07FE                         190           BR     14
                                    191 **K
0001E8                              192 AUSD1     DS     0F
0001EP 40E2E3C1E3C9E2E3             193           DC     CL24' STATISTIK DES ZUFALLSGE'
000200 D5C5D9C1E3D6D9E2             194           DC     CL8'NERATORS'
000208                              195 AUSD2     DS     0F
000208 5C5C5C5C5C5C5C5C             196           DC     32CL1'*'
000228                              197 AUSD3     DS     0F
000228 40404040                     198           DC     4CL1' '
00022C                              199 AUSD4     DS     0F
00022C                              200           DS     F
000230                              201 AUSD8     DS     0F
000230 4040404040404040             202           DC     8CL1' '
000238                              203 AUSD5     DS     0F
000238                              204           DS     2F
000240                              205 AUSD7     DS     0F
000240 F0F1F2F3F4F5F6F7             206           DC     CL10'0123456789'
                                    207 **K
                                    208 **K
000250                              209           LTORG
00025C 00010005                     210                  =F'65541'
000254 0000000A                     211                  =F'10'
000258 00000005                     212                  =F'5'
00025C 00000004                     213                  =F'4'
00026C 00000001                     214                  =F'1'
000264 00000014                     215                  =F'20'
000268 00000[illegible]             216                  =V(OUTPUT@)
00026C 0000[illegible]              217                  =F'32'
000270 00000010                     218                  =V(AUSW3)
000274 00000000                     219                  =F'0'
000278 00000009                     220                  =F'9'
00027C 00000008                     221                  =F'8'
000280 00000000                     222                  =V(BD@)
000284 40404040                     223                  =CL4'    '
000288 00000010                     224                  =F'16'
                                    225 **K
                                    226 **K
                                    227 **K  HAUPTPROGRAMM
                                    228 **K
000290                              229 HAUPT     CSECT
000290                              230           USING *,15
                                    231 **K
                                    232 **K  SIND DIE ZAHLEN ZU DRUCKEN?
                                    233 **K
000290 5810 F1B0             00440  234           L      1,=F'1'
000294 4100 F158             003E8  235           LA     0,HAUPT1
000298 58B0 F1B4             00444  236           L      11,=F'28'
00029C 18CF                         237           LR     12,15
00029E 58F0 F1B8             00448  238           L      15,=V(OUTPUT@)
0002A2 05EF                         239           BALR   14,15
0002A4 18FC                         240           LR     15,12
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                    F01MAY72  12/12/73

0002AE 5810 F1BC            0044C   241           L     1,=F'4'
0002B2 4100 F174            00404   242           LA    0,HAUPT2
0002B6 58B0 F1BC            0044C   243           L     11,=F'4'
0002BA 18CF                         244           LR    12,15
0002BC 58F0 F1B8            00448   245           L     15,=V(OUTPUT@)
0002C0 05EF                         246           BALR  14,15
0002C2 18FC                         247           LR    15,12
                                    248 **K
                                    249 **K  LIES ANZAHL DER ZAHLEN EIN
                                    250 **K
0002C4 5810 F1B0            00440   251           L     1,=F'1'
0002C8 4100 F178            00408   252           LA    0,HAUPT4
0002CC 58B0 F1C0            00450   253           L     11,=F'36'
0002D0 18CF                         254           LR    12,15
0002D2 58F0 F1B8            00448   255           L     15,=V(OUTPUT@)
0002D6 05EF                         256           BALR  14,15
0002D8 18FC                         257           LR    15,12
0002DA 5810 F1BC            0044C   258           L     1,=F'4'
0002DE 41B0 F150            003F0   259           LA    11,BUFFER
0002E2 41BB 0004            00004   260           LA    11,4(11)
0002E6 180B                         261           LR    0,11
0002E8 58B0 F1BC            0044C   262           L     11,=F'4'
0002EC 18CF                         263           LR    12,15
0002EE 58F0 F1B8            00448   264           L     15,=V(OUTPUT@)
0002F2 05EF                         265           BALR  14,15
0002F4 18FC                         266           LR    15,12
0002F6 5800 F1C4            00454   267           L     0,=CL4'0000'
0002FA 5000 F150            003E0   268           ST    0,BUFFER
0002FE 4110 F150            003E0   269           LA    1,BUFFER
000302 5800 F1C8            00458   270           L     0,=F'8'
000306 18CF                         271           LR    12,15
000308 58F0 F1CC            0045C   272           L     15,=V(DB@)
00030C 05EF                         273           BALR  14,15
00030E 18FC                         274           LR    15,12
000310 5000 F150            003E0   275           ST    0,BUFFER
                                    276 **K
                                    277 **K  AUSFUEHRUNGSSCHLEIFE
                                    278 **K
000314 5830 F1B0            00440   279           L     3,=F'1'
000318 5840 F1B0            00440   280           L     4,=F'1'
00031C 5850 F150            003E0   281           L     5,BUFFER
000320                              282 L@0004    EQU   *
000320 1935                         283           CR    3,5
000322 4720 F130            003C0   284           BH    L@0005
000326 50FD 0000            00000   285           ST    15,0(13)
00032A 41DD 0004            00004   286           LA    13,4(13)
00032E 5810 F1D0            00460   287           L     1,=V(RANDOM)
000332 58F0 F1D4            00464   288           L     15,=V(ROUTINEN)
000336 05E1                         289           BALR  14,1
000338 4100 0004            00004   290           LA    0,4
00033C 1BD0                         291           SR    13,0
00033E 58FD 0000            00000   292           L     15,0(13)
000342 5020 F150            003E0   293           ST    2,BUFFER
000346 5890 F1C8            00458   294           L     9,=F'8'
00034A 5810 F150            003E0   295           L     1,BUFFER
00034E 1809                         296           LR    0,9
000350 41B0 F150            003E0   297           LA    11,BUFFER
000354 18CF                         298           LR    12,15
000356 58F0 F1D8            00468   299           L     15,=V(BD@)
00035A 05EF                         300           BALR  14,15
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                  F01MAY72  12/12/73

000354 18FC                          301           LR    15,12
000356 5860 F1DC              0046C   302           L     6,=F'0'
00035A 5870 F1B0              00440   303           L     7,=F'1'
00035E 5880 F1E0              00470   304           L     8,=F'6'
000362                                305 L@0006    EQU   *
000362 1968                           306           CR    6,8
000364 4720 F0E6              00376   307           BH    L@0007
000368 4300 F1EC              0047C   308           IC    0,=CL1' '
00036C 4206 F150              003E0   309           STC   0,BUFFER(6)
000370 1A67                           310           AR    6,7
000372 47F0 F0D2              00362   311           B     L@0006
000376                                312 L@0007    EQU   *
00037E 5800 F174              004C4   313           L     0,HAUPT2
00037A 5810 F19C              0042C   314           L     1,HAUPT5
00037E 1901                           315           CR    0,1
000380 4770 F0F8              00388   316           BNE   L@0008
000384 47F0 F10E              0039E   317           B     HAUPT3
000388                                318 L@0008    EQU   *
000388 5810 F1B0              00440   319           L     1,=F'1'
00038C 4100 F150              003E0   320           LA    0,BUFFER
000390 5880 F1C8              00458   321           L     11,=F'8'
000394 18CF                           322           LR    12,15
000396 58F0 F1B8              00448   323           L     15,=V(OUTPUT@)
00039A 05EF                           324           BALR  14,15
00039C 18FC                           325           LR    15,12
00039E                                326 HAUPT3    EQU   *
00039E 50FD 0000              000C0   327           ST    15,0(13)
0003A2 41DD 0004              000C4   328           LA    13,4(13)
0003A6 5810 F1E4              00474   329           L     1,=V(AUSWERT)
0003AA 58F0 F1D4              00464   330           L     15,=V(RCUTINEN)
0003AE 05E1                           331           BALR  14,1
0003B0 4100 0004              000C4   332           LA    0,4
0003B4 1BD0                           333           SR    13,0
0003B6 58FD 0000              000C0   334           L     15,0(13)
0003BA 1A34                           335           AR    3,4
0003BC 47F0 F088              00318   336           B     L@0004
0003C0                                337 L@0005    EQU   *
                                      338 **K
                                      339 **K  DRUCKE DIE STATISTIK
                                      340 **K
0003CC 50FD 0000              00000   341           ST    15,0(13)
0003C4 41DD 0004              00004   342           LA    13,4(13)
0003C8 5810 F1E8              00478   343           L     1,=V(AUSDRUK)
0003CC 58F0 F1D4              00464   344           L     15,=V(ROUTINEN)
0003D0 05E1                           345           BALR  14,1
0003D2 4100 0004              00004   346           LA    0,4
0003D6 1BDC                           347           SR    13,0
0003D8 58FD 0000              000C0   348           L     15,0(13)
0003DC 47F0 F1A0              00430   349           B     END
                                      350 **K
0003E0                                351 BUFFER    DS    0F
0003E0                                352           DS    2F
0003E8                                353 HAUPT1    DS    0F
0003E8 40E2C9D5C440C4C9               354           DC    CL28' SIND DIE ZAHLEN ZU DRUCKEN?'
000404                                355 HAUPT2    DS    0F
000404                                356           DS    CL4
000408                                357 HAUPT4    DS    0F
000408 40E6C9C5E5C9C5D3               358           DC    CL16' WIEVIELE ZAHLEN'
000418 40E2C9D5C440E9E4               359           DC    CL20' SIND ZU GENERIEREN?'
00042C                                360 HAUPT5    DS    0F
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT
000420 D5C5C9D5                     361          DC     CL4'NEIN'
                                    362 **K
                                    363 **K
00043C                              364 END      EQU    *
000430 58D0 F1BC             0044C  365          S      13,=F'4'
000434 58DD 0000             00000  366          L      13,0(13)
000438 98EC D00C             0000C  367          LM     14,12,12(13)
00043C C7FE                         368          BR     14
000440                              369          LTORG
000440 00000001                     370                 =F'1'
000444 0000001C                     371                 =F'28'
000448 00000000                     372                 =V(OUTPUT@)
00044C 0000C004                     373                 =F'4'
000450 00000024                     374                 =F'36'
000454 F0F0F0F0                     375                 =CL4'0000'
000458 00000008                     376                 =F'8'
00045C 00000000                     377                 =V(DB@)
000460 00000000                     378                 =V(RANDOM)
000464 00000000                     379                 =V(ROUTINEN)
000468 0000C000                     380                 =V(RD@)
00046C 00000000                     381                 =F'0'
000470 0000C006                     382                 =F'6'
000474 00000000                     383                 =V(AUSWERT)
000478 00000000                     384                 =V(AUSDRUK)
00047C 40                           385                 =CL1' '
000480                              386 SAVEPT@  CSECT
000480 00000484                     387          DC     A(*+4)
000484                              388          DS     100F
000000                              389          END    ZUFALL
```

RELOCATION DICTIONARY

12/12/73

POS.ID	REL.ID	FLAGS	ADDRESS
01	02	1C	000030
01	03	1C	000034
04	05	1C	000268
04	06	1C	000270
04	07	1C	000280
08	05	1C	000448
08	07	1C	000468
08	09	1C	00045C
08	0A	1C	000460
08	0B	1C	000464
08	0C	1C	000474
08	0D	1C	000478
0F	0E	0C	000480

CROSS-REFERENCE PAGE 1

12/12/73

SYMBOL	LEN	VALUE	DEFN	REFERENCES									
AUSDRUK	00001	0000F8	00112	0035									
AUSD1	00004	0001E8	00192	0130									
AUSD2	00004	000208	00195	0137									
AUSD3	00004	000228	00197	0123	0144								
AUSD4	00004	00022C	00199	0162	0163								
AUSD5	00004	000238	00203	0165	0171								
AUSD7	00004	000240	00205	0172									
AUSD8	00004	000230	00201	0173	0175								
AUSWERT	00001	00006C	00074	0033									
AUSW1	00004	000080	00102	0082									
AUSW2	00001	000080	00097	0088									
AUSW3	00004	000000	00104	0034	0093	0095							
BUFFER	00004	0003E0	00351	0259	0268	0269	0275	0281	0293	0295	0297	0309	0320
END	00001	000430	00364	0349									
HAUPT	00001	000290	00229										
HAUPT1	00004	0003E8	00353	0235									
HAUPT2	00004	000404	00355	0242	0313								
HAUPT3	00001	00039E	00326	0317									
HAUPT4	00004	000408	00357	0252									
HAUPT5	00004	00042C	00360	0314									
L@0000	00001	00009E	00090	0080									
L@0001	00001	000080	00096	0089									
L@0002	00001	000176	00156	0182									
L@0003	00001	0001CC	00183	0158									
L@0004	00001	000318	00282	0336									
L@0005	00001	0003C0	00337	0284									
L@0006	00001	000362	00305	0311									
L@0007	00001	000376	00312	0307									
L@0008	00001	000388	00318	0316									
PROLOG	00001	000024	00022	0019									
RANDOM	00001	000038	00047	0032									
RAND1	00004	000068	00065	0050	0054								
ROUTINEN	00001	000038	00030										
SAVEPT@	00001	000480	00386										
ZUFALL	00001	000000	00008	0389									

NO STATEMENTS FLAGGED IN THIS ASSEMBLY
STATISTICS SOURCE RECORDS (SYSIN) = 356
OPTIONS IN EFFECT LIST, NODECK, LOAD, NORENT, XREF, NOTEST, ALGN, OS, NOTERM, LINECNT = 60
481 PRINTED LINES

```
F88-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED MAP,LIST,CALL
          DEFAULT OPTION(S) USED -  SIZE=(114688,32768)
IEW0000      INCLUDE ORJ(FICLFZT)                                    00000310
```

MODULE MAP

CONTROL SECTION			ENTRY							
NAME	ORIGIN	LENGTH	NAME	LOCATION	NAME	LOCATION	NAME	LOCATION	NAME	LOCATION
ZUFALL	00	38								
			PROLOG	24						
ROUTINEN	38	254								
			RANDOM	38	AUSWERT	6C	AUSW3	D0	AUSDRUK	F8
HAUPT	290	1ED								
SAVEPT@	480	194								
LAUFZEIT	618	30C								
			OUTPUT@	618	DB@	874	BD@	89E		

```
ENTRY ADDRESS       00
TOTAL LENGTH       928

****MAIN      DOES NOT EXIST BUT HAS BEEN ADDED TO DATA SET
```

```
**K
**K
**K   EIN BEISPIELPROGRAMM IN ASR
**K
**K   PSEUDO-ZUFALLSZAHLENGENERATOR
**K
**K
          IDENT     ZUFALL
          ENTRY     ZUFALL
          EXT       OUTPUT.,BD.,DB.
          COMMON
R15       BSS       1
R14       BSS       1
R13       BSS       1
R12       BSS       1
R11       BSS       1
R10       BSS       1
R9        BSS       1
R8        BSS       1
R7        BSS       1
R6        BSS       1
R5        BSS       1
R4        BSS       1
R3        BSS       1
R2        BSS       1
R1        BSS       1
R0        BSS       1
SAVEPT.   BSS       1
          BSS       100
          PRG
ZUFALL    UJP       **
          ENA       SAVEPT.
          INA       1
          STA       SAVEPT.
**K
          UJP       HAUPT
**K
**K
**K
**K   ZUFALLSZAHLENGENERATOR NACH DER
**K   MULTIPLIKATIVEN KONGRUENZMETHODE
**K
**K   MULTIPLIKAND 2**16+5 (65541)
**K   DIVISOR 2**32
**K
**K   ZUFALLSZAHL ZWISCHEN 0 UND 9 WIRD
**K   IN REG. 2 UEBERGEBEN
**K
RANDOM    EQU       *
          UJP       **
          LDA       *-1
          STA,I     SAVEPT.
          ENA       1
          RAD       SAVEPT.
          LDA       RAND1
          STA       R2
          MUA       =D65541
          STA       R2
          STA       RAND1
```

```
**K
          LDA       R2
          MUA       =D10
          STA       R2
          LDA       R0
          INA       5
          STA       R0
          STA       R2
          ENA,S     -1
          RAD       SAVEPT.
          LDI       SAVEPT.,1
          UJP,I     0,1
**K
RAND1     BSS       0
          DEC       8193
**K
**K
**K  ROUTINE ZUR AUSWERTUNG DES
**K  ZUFALLSZAHLENGENERATORS
**K
**K  ZUFALLSZAHL IST IN REG. 2
**K
AUSWERT   EQU       *
          UJP       **
          LDA       *-1
          STA,I     SAVEPT.
          ENA       1
          RAD       SAVEPT.
          ENQ       10
          LDA       R2
          AQJ,LT    L.0000
          ENA       1
          STA       R1
          ENA       AUSW1
          STA       R0
          ENA       20
          STA       R11
          RTJ       OUTPUT.
          UJP       AUSW2
          UJP       L.0001
L.0000    EQU       *
          LDI       R2,1
          LDA       AUSW3,1
          INA       1
          LDI       R2,1
          STA       AUSW3,1
L.0001    EQU       *
AUSW2     EQU       *
          ENA,S     -1
          RAD       SAVEPT.
          LDI       SAVEPT.,1
          UJP,I     0,1
**K
AUSW1     BSS       0
          BCD,C     20, FEHLER IN RANDOM
AUSW3     BSS       0
          DEC       0
          DEC       0
          DEC       0
```

```
          DEC       0
          DEC       0
          DEC       0
          DEC       0
          DEC       0
          DEC       0
          DEC       0
**K
**K
**K  ROUTINE ZUM AUSDRUCKEN DER STATISTIK
**K
**K  TABELLE STEHT BEI ADRESSE AUSW3
**K
AUSDRUK   EQU       *
          UJP       **
          LDA       *-1
          LDI       SAVEPT.,1
          STA       0,1
          LDA       R2
          STA       1,1
          LDA       R3
          STA       2,1
          LDA       R4
          STA       3,1
          LDA       R5
          STA       4,1
          ENA       4
          RAD       SAVEPT.
**K
**K  DRUCKE TITELZEILE
**K
          ENA       1
          STA       R1
          ENA       AUSD3
          STA       R0
          ENA       4
          STA       R11
          RTJ       OUTPUT.
          ENA       1
          STA       R1
          ENA       AUSD1
          STA       R0
          ENA       32
          STA       R11
          RTJ       OUTPUT.
          ENA       1
          STA       R1
          ENA       AUSD2
          STA       R0
          ENA       32
          STA       R11
          RTJ       OUTPUT.
          ENA       1
          STA       R1
          ENA       AUSD3
          STA       R0
          ENA       4
          STA       R11
          RTJ       OUTPUT.
```

```
**K
**K   DRUCKSCHLEIFE
**K
          ENA       AUSW3
          STA       R5
          ENA       0
          STA       R2
          ENA       1
          STA       R3
          ENA       9
          STA       R4
L.0002    EQU       *
          LDQ       R4
          LDA       R2
          AQJ,GE    L.0003
          LDI       R2,1
          LDA       R5
          SWA       *+1
          LDA       **,1
          STA       AUSD4
          STA       R1
          ENA       8
          STA       R0
          ENA       AUSD5
          STA       R11
          RTJ       BD.
          LDA       =H
          STA       AUSD5
          LDI       R2,1
          LACH      AUSD7,1
          SACH      AUSD8+3
          ENA       1
          STA       R1
          ENA       AUSD8
          STA       R0
          ENA       16
          STA       R11
          RTJ       OUTPUT.
          LDA       R2
          ADA       R3
          STA       R2
          UJP       L.0002
L.0003    EQU       *
          ENA,S     -4
          RAD       SAVEPT.
          LDI       SAVEPT.,1
          LDA       1,1
          STA       R2
          LDA       2,1
          STA       R3
          LDA       3,1
          STA       R4
          LDA       4,1
          STA       R5
          UJP,I     0,1
**K
AUSD1     BSS       0
          BCD,C     24, STATISTIK DES ZUFALLSGE
          BCD,C     8,NERATORS
```

```
AUSD2     BSS       0
          BCD,C     32,********************************
AUSD3     BSS       0
          BCD,C     4,
AUSD4     BSS       0
          BSS       1
AUSD8     BSS       0
          BCD,C     8,
AUSD5     BSS       0
          BSS       2
AUSD7     BSS       0
          BCD,C     10,0123456789
**K
**K
**K
**K
**K  HAUPTPROGRAMM
**K
HAUPT     EQU       *
**K
**K  SIND DIE ZAHLEN ZU DRUCKEN?
**K
          ENA       1
          STA       R1
          ENA       HAUPT1
          STA       R0
          ENA       28
          STA       R11
          RTJ       OUTPUT.
          ENA       4
          STA       R1
          ENA       HAUPT2
          STA       R0
          ENA       4
          STA       R11
          RTJ       OUTPUT.
**K
**K  LIES ANZAHL DER ZAHLEN EIN
**K
          ENA       1
          STA       R1
          ENA       HAUPT4
          STA       R0
          ENA       36
          STA       R11
          RTJ       OUTPUT.
          ENA       4
          STA       R1
          ENA       BUFFER
          INA       1
          STA       R0
          ENA       4
          STA       R11
          RTJ       OUTPUT.
          LDA       =H0000
          STA       BUFFER
          ENA       BUFFER
          STA       R1
          ENA       8
```

```
          STA       R0
          RTJ       DB.
          LDA       R0
          STA       BUFFER
**K
**K   AUSFUEHRUNGSSCHLEIFE
**K
          ENA       1
          STA       R3
          ENA       1
          STA       R4
          LDA       BUFFER
          STA       R5
L.0004    EQU       *
          LDQ       R5
          LDA       R3
          AQJ,GE    L.0005
          EXT       RANDOM
          RTJ       RANDOM
          LDA       R2
          STA       BUFFER
          ENA       8
          STA       R9
          LDA       BUFFER
          STA       R1
          LDA       R9
          STA       R0
          ENA       BUFFER
          STA       R11
          RTJ       BD.
          ENA       0
          STA       R6
          ENA       1
          STA       R7
          ENA       6
          STA       R8
L.0006    EQU       *
          LDQ       R8
          LDA       R6
          AQJ,GE    L.0007
          LACH      =H
          LDI       R6,2
          SACH      BUFFER,2
          LDA       R6
          ADA       R7
          STA       R6
          UJP       L.0006
L.0007    EQU       *
          LDQ       HAUPT5
          LDA       HAUPT2
          AQJ,NE    L.0008
          UJP       HAUPT3
L.0008    EQU       *
          ENA       1
          STA       R1
          ENA       BUFFER
          STA       R0
          ENA       8
          STA       R11
```

```
          RTJ       OUTPUT.
HAUPT3    EQU       *
          EXT       AUSWERT
          RTJ       AUSWERT
          LDA       R3
          ADA       R4
          STA       R3
          UJP       L.0004
L.0005    EQU       *
**K
**K  DRUCKE DIE STATISTIK
**K
          EXT       AUSDRUK
          RTJ       AUSDRUK
          UJP       END
**K
BUFFER    BSS       0
          BSS       2
HAUPT1    BSS       0
          BCD,C     28, SIND DIE ZAHLEN ZU DRUCKEN?
HAUPT2    BSS       0
          BSS,C     4
HAUPT4    BSS       0
          BCD,C     16, WIEVIELE ZAHLEN
          BCD,C     20, SIND ZU GENERIEREN?
HAUPT5    BSS       0
          BCD,C     4,NEIN
**K
**K
END       EQU       *
          UJP,I     ZUFALL
          END       ZUFALL
          FINIS
```