Ein tabellengesteuerter Kommandoentschlüßler für generierbare Kommandosyntax

M. Rupp

Gesellschaft für Kernforschung Karlsruhe,
Institut für Datenverarbeitung in der Technik

Ein tabellengesteuerter Kommandoentschlüßler für generierbare Kommandosyntax

Martin Rupp

Institut für Datenverarbeitung in der Technik
Kernforschungszentrum Karlsruhe

1. Einleitung

Der vorliegende Bericht beschreibt ein allgemein anwendbares Verfahren zur Entschlüsselung von Kommandos, die einer generierbaren Kommandosyntax genügen.
Die Entschlüsselung eines Kommandowortes wird unabhängig von der kommandoausführenden Funktion durch kommandobeschreibende Tabellen gesteuert. Die Tabellen ermöglichen die Verwendung der unterschiedlichsten Kommandosprachen und können installationsabhängig generiert werden.
Das Verfahren ist für den Einsatz auf Kleinrechnern zugeschnitten und wird beim MZS 330 angewandt.

2. Ablauf der Kommandoentschlüsselung

Kommunikationsmittel zwischen Kommandosystem und Anwender ist das Kommandowort, das sich aus dem Kommandonamen und gegebenenfalls einer Operandenfolge zusammensetzt. Der Kommandoname dient zur Auswahl der gewünschten Funktion, die Operandenfolge - die einer Operandensyntax genügen muß - zur Spezifikation der Funktionsparameter.
Bild 1 zeigt eine Organisationsform der Kommandoentschlüsselung, die man bei Kommandosystemen auf kleineren Rechenanlagen häufig vorfindet. Ein auf das System zugeschnittener Kommandonamensentschlüßler dekodiert den Kommandonamen und ruft die kommandoausführende Funktion auf, die in eigener Regie die Operandenfolge entschlüsselt.

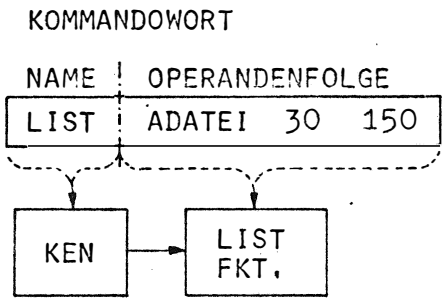
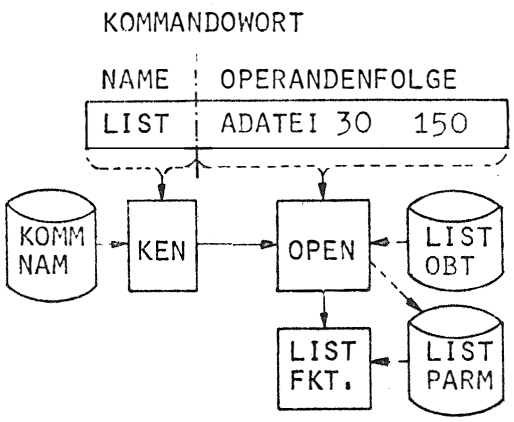


BILD 1 FUNKTIONSABHÄNGIGE ENTSCHLÜSSELUNG

Nachteile einer solchen Organisation sind
- beim Erstellen des Systems
  - hoher Dekodierungsaufwand je Funktion
  - erschwerte Schnittstellendefinition bei einer Aufgabenteilung
- im Betrieb
  - Eingriffe in den Ablaufcode bei Umformungen der Kommandosprache.

Bild 2 verdeutlicht den Ablauf des tabellengesteuerten Kommandoentschlüßlers als zweistufigen Vorgang von
- Kommandonamenserkennung und
- Operandenentschlüsselung.

In der ersten Stufe erfolgt die Dekodierung des vorliegenden Kommandonamens anhand einer Liste, in der die zulässigen Kommandonamen verzeichnet sind. In der zweiten Stufe wird die Operandenfolge mit Hilfe der Operandenbeschreibungstabelle entschlüsselt, wobei die erhaltenen Werte in den funktionsspezifischen Parameterblock geschrieben werden. Die Operandenbeschreibungstabelle beinhaltet neben der Abarbeitungsvorschrift der Operandenfolge die Bezüge zwischen Operanden und Ablage-Art bzw. -Platz im Parameterblock. Der Parameterblock ist somit Schnittstelle zwischen Operandenentschlüsselung und kommandoausführender Funktion.



KOMMNAM: KOMMANDONAMENSLISTE
OBT : OPERANDENBESCHREIBUNGS-TABELLE
KEN : KOMMANDONAMENSENTSCHL.
OPEN : OPERANDENENTSCHLÜSSELER

BILD 2 FUNKTIONSLOSGELÖSTE ENTSCHLÜSSELUNG

Vorteile dieser Organisationsform sind
- beim Erstellen des Systems
  - modulare Trennung von Kommandonamenserkennung, Operandenentschlüsselung und kommandoausführender Funktion
  - Speicherbedarf von weniger als 1 K Worte

- im Betrieb
  - geringer Aufwand zur Eingliederung neuer Funktionen in das System
  - Generierbarkeit anwendungsangepaßter Kommandosprachen.

3. Realisierung

Im folgenden wird vorausgesetzt, daß ein Zeichen maschinenintern in einem Byte (8 Bit) abgelegt wird.

3.1. Kommandonamenserkennung

Zur Kommandonamenserkennung wird als speicherplatzminimales Suchverfahren das sequentielle Durchsuchen einer gepackten Liste verwendet.
Bild 3 zeigt die Darstellung eines Kommandonamens in der Liste.

Bei der Systemgenerierung wird neben dem Kommandonamen und dem das Kommando bezeichnenden Schlüssel noch eine Abkürzungslänge eingegeben, die ausdrückt, wieviele Zeichen des Namensanfangs den Namen eindeutig kennzeichnen.

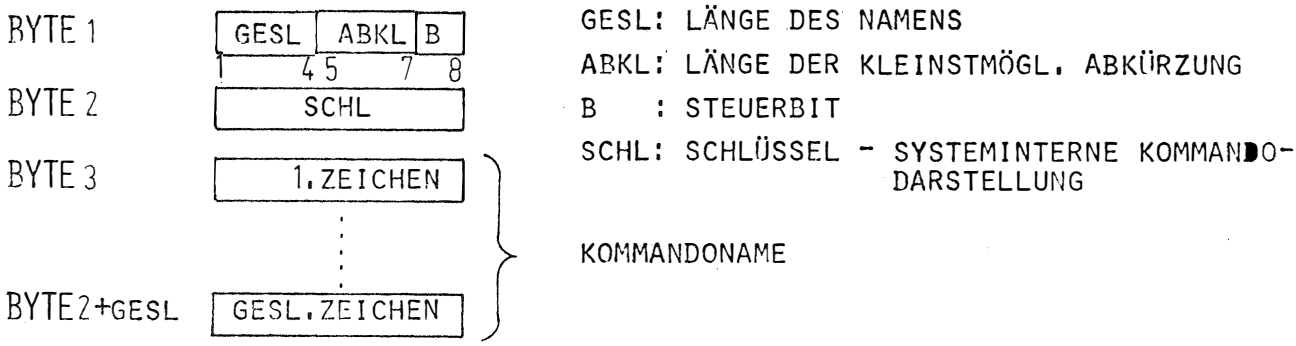


BILD 3 DARSTELLUNG EINES NAMENS IN DER KOMMANDONAMENSLISTE

3.2. Operandenentschlüsselung

Die wesentliche Aufgabe der Kommandoentschlüsselung besteht in der Entschlüsselung der Operandenfolge.
Der vorliegende Operandenentschlüßler unterscheidet die Operanden einer Operandenfolge aufgrund
- ihres Typs
- ihrer Bezeichnung
- ihrer Anordnung.

3.2.1. Operandentypen

Die Spezifizierung der Operandentypen setzt die Aufteilung der vorhandenen Zeichenmenge in die disjunkten Mengen: Ziffern, Buchstaben, Trennzeichen, Sonderzeichen und Restzeichen voraus.

Die zulässigen Operandentypen, ihre Bedeutung und ihr Bezug zum Parameterblock sind im Bild 4 tabellarisch beschrieben.

TYP	DARSTELLUNG U. BEDEUTUNG	PARAMETERBLOCK-EINTRAG	BEISPIELE
ZAHL	IN EINEM MASCHINENWORT DARSTELLBARE POSITIVE GANZE ZAHL	NUMERISCHER WERT VON ZAHL	5 1034
WORT	TRENNZEICHENFREIE ZEICHENFOLGE MIT FUEHRENDEM BUCHSTABEN	ZEICHENFOLGE VON WORT	ABC1 A3/B
ZFOLGE	DURCH BELIEBIGES SONDERZEICHEN EINGESCHLOSSENE ZEICHENFOLGE	BEIDE ZEICHENPUFFERADRESSEN DES SONDERZEICHENS	/A 1.B+/ & D/5 = &
KENNWORT	VORGEGEBENES WORT ALS SYNONYM EINES KOMM.-SPEZIFISCHEN WERTES	SCHLUESSEL VON KENNWORT	ALL NUM1

BILD 4 OPERANDEN-TYPEN

3.2.2. Operandentrennung

Die Operanden werden gegenüber dem Kommandonamen und untereinander entweder durch ein oder mehrere Leerzeichen oder durch ein Trennzeichen - umgeben von beliebig vielen Leerzeichen - getrennt (Bsp.: op1,op2,op3 ; op4).

3.2.3. Operandenbezeichnung

Zur Differenzierung oder Verdeutlichung kann einem Operanden ein BEZEICHNER zugeordnet werden. Ein bezeichneter Operand ist nur in unmittelbarer Folge auf seinen BEZEICHNER angebbar.

Die BEZEICHNER werden bei der Operandendefinition festgelegt und sind formgleich mit dem Operandentyp KENNWORT. Das letzte Zeichen eines BEZEICHNERs gilt als Operandentrennzeichen. Die Zeichenfolge davor kann vereinbarungsgemäß abgekürzt werden (Bsp.: ART=op bzw. A=op).

KENNWORT und BEZEICHNER werden in der Operandenbeschreibungstabelle nach dem gleichen Verfahren angelegt und erkannt wie der Kommandoname.

Die Operandenerkennung aufgrund der Anordnungsfolge wird durch die Struktur der Operandenbeschreibungstabelle gewährleistet.

3.2.4. Operandenbeschreibungstabelle

Die Operandenbeschreibungstabelle beinhaltet neben den KENNWORT- und BEZEICHNER-Listen sogenannte Organisationselemente, die den Ablauf der Entschlüsselung steuern.
Folgende sechs Typen von Organisationselementen werden benötigt:

Typ 1 beschreibt einen ZAHL-Operanden
Typ 2 " " WORT-Operanden
Typ 3 " " ZFOLGE-Operanden
Typ 4 " " KENNWORT-Operanden
Typ 5 " " BEZEICHNER und den zugehörigen Operanden
Typ 6 ist Begrenzungselement.

Neben der Beschreibung der Operanden (bei Typ 1 - 5) enthält jedes Element die zwei Element-Zeiger ODER und NEXT.

3.2.5. Strategie des Operandenentschlüßlers

Die Operandenentschlüsselung beginnt mit dem Vergleich der ersten durch Trennzeichen getrennten Teilfolge der zu analysierenden Zeichenfolge mit dem Operandentyp des Ansatz-

(ersten) Organisationselementes. Bei Übereinstimmung wird der ermittelte Wert in den Parameterblock eingetragen und die nächste Teilfolge mit dem Operandentyp des Elementes verglichen, auf das NEXT verweist. Entspricht die Teilfolge nicht dem Typ des aktuellen Elementes, so wird der Vergleich bei dem durch ODER spezifizierten, alternativen Element fortgesetzt. Fehlt die Angabe einer Alternative, so wird die Operandenentschlüsselung mit einer entsprechenden Fehlermeldung abgebrochen. Die Entschlüsselung gilt als erfolgreich beendet, wenn alle Zeichen analysiert sind, und der aktuelle Wert von NEXT ein Begrenzungselement bestimmt.
Daneben dient das Begrenzungselement bei gezielter Anwendung von NEXT und ODER zur

- Verwendung stellungsfreier Operanden
- Unterbrechung der Entschlüsselung (im Hinblick auf die Wiederaufnahme bei dem durch NEXT bestimmten Organisationselement).

3.2.6. Beispiel

Das vorliegende Beispiel soll demonstrieren, wie mit Hilfe der Organisationselemente selbst eine komplexe Operandensyntax durch eine einfache Datenstruktur ausdrückbar ist.
Es gelte folgende Nomenklatur:

- Operanden sind in Kleinbuchstaben, BEZEICHNER in Großbuchstaben dargestellt
- <op1> : die Angabe von op1 ist optional
- (op1,op2): op1 und op2 sind stellungsfrei und optional
- {op1 / op2} : op1 und op2 sind Alternativen
- [op1] : op1 ist wiederholt angebbar.

Die Operandenfolge genüge folgender Operandensyntax:

op1 <op2> { op3 ({VON: / FROM=} op4 , {BIS: / TO=} op5) / [op6] }

mit: op1, op2 : ZAHL
op3 : KENNWORT
op4, op5 : ZFOLGE
op6 : WORT

Richtige Operandenfolgen sind:
- op1 op3 BIS:op5 FROM=op4
- op1 op2 op3
- op1 op6 op6 op6

Die resultierende Datenstruktur ist in vereinfachter Form (Beschreibungen der Operanden fehlen) im Bild 5 dargestellt.

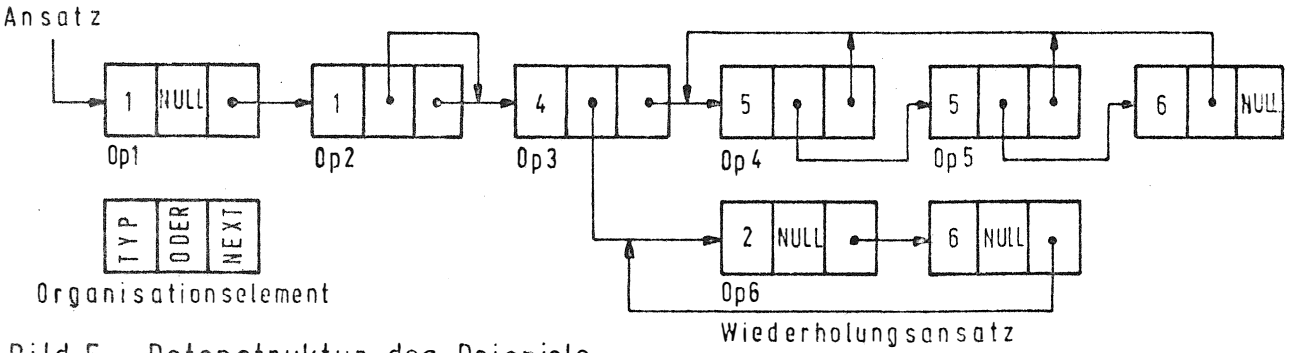


Bild 5 Datenstruktur des Beispiels

4. Schlußfolgerung

Die Benutzung des Verfahrens erfordert eine genaue Kenntnis über Form und Aufbau der Tabellen, sowie über die Abarbeitungsregeln des Kommando- und Operandenentschlüßlers.
Die Struktur der Operandenbeschreibungstabelle ermöglicht vielfältige Darstellungen einer Operandensyntax und damit die Verwendung einer komfortablen Kommandosprache.
Die Anwendung des Verfahrens innerhalb eines Kommandosystems ermöglicht dem
- Systemhersteller
  - die Konzipierung der kommandoausführenden Funktionen ohne Rücksicht auf die Art und Weise der Parameterversorgung
  - eine erleichterte Arbeitsaufteilung aufgrund klarer Schnittstellen
- Systeminstallateur
  - die Anpassung der Kommandosprache an die Wünsche und das Sprachverständnis des Benutzerkreises
  - das Sperren von Parametern zum Zwecke der Anpassung der Funktionen an den Systemausbau oder zur Verhinderung einer mißbräuchlichen Anwendung.