

Implementierter PEARL - SprachumfangPEARL - CompilerPORTABLES PEARL - PROGRAMMIERSYSTEM(COMPILER, TESTSYSTEM, DATENBANKSYSTEM)

Mittels solcher Compiler-Parameter können u.a. Listings der Quell- und Zielprogramme erzeugt werden, wobei im Assembler- oder Bindcode-Format die Ursprungszeilen im Quellprogramm eingefügt sind. Auf Wunsch eine Cross-Referenz-Liste aller im Quellprogramm verwendeten Variablen, Operatoren, Prozeduren, Funktionen, etc. mit ihren Nummern für ihre Definition und Referenz. Durch Compiler-Optionen kann die Erzeugung von Feldgrenzen und Referenzen zur Laufzeit unterdrückt werden. Der Compiler analysiert Programme um Fehler zu erkennen und meldet sie im Klartext mit Angabe der Quellzeilen-Nummer. Verschiedene Fehlermeldungen sind möglich.

Ein Präprozessor gestattet das Einfügen von Programmteilen aus anderen Dateien. Das portable PEARL-Programmiersystem von WERUM besitzt folgende Komponenten:

- als System-Auswahl wird durch eine Konfigurationstabelle gesteuert, welche Compiler, Testsystem, Datenbank, etc. für die jeweilige Plattform ausgewählt werden.
- Compiler (Rechnerunabhängiges, einmal erstelltes Objekt und rechnerabhängig zu erzeugendes Binärformat, bestehend aus einer Reihe von Binärformaten, z.B. IBM, etc., die der Anwender selbst mit einfachen Mitteln in eine Konfigurationstabelle eintragen kann).
- Betriebssystem-Kern
- Laufzeit-System, bestehend aus:
 - Laufzeit-Objekt für binäre E/A
 - Laufzeit-Objekt für formatierte E/A
 - Testsystem
 - Datenbanksystem

HANS WINDAUER

ENTWICKLUNGSBÜRO WULF WERUM

2120 LÜNEBURG

Nur einmalige Berechnung der Adressen von Variablen und Feldern ist notwendig. Die Laufzeit-Systeme sind rechnerunabhängig und werden bei den bisherigen Implementierungen jeweils für die jeweilige Plattform separat erstellt. Die Laufzeit-Systeme sind rechnerunabhängig und werden bei den bisherigen Implementierungen jeweils für die jeweilige Plattform separat erstellt. Die Laufzeit-Systeme sind rechnerunabhängig und werden bei den bisherigen Implementierungen jeweils für die jeweilige Plattform separat erstellt.

Im folgenden werden die Komponenten Compiler, Testsystem und Datenbanksystem näher vorgestellt.

Der Compiler ist in GBL, einem echten PL/I-Subset, programmiert und kann somit einfach auf Rechnern installiert werden, die über einen PL/I-Compiler verfügen (als Cross-Compiler). Auf Rechnern, die über einen PL/I-Compiler verfügen, kann der Compiler auch als Cross-Compiler installiert werden.

Implementierter PEARL - Sprachumfang

Der PEARL-Compiler von WERUM verarbeitet Programme laut DIN 66253 (Basic PEARL, Teil 1, Entwurf) sowie weitere Sprachelemente aus Full PEARL, die für eine anwendungsorientierte und effiziente Programmierung wesentlich sind. Die wichtigsten dieser Sprachelemente sind:

- Referenzen (REF)
- Neu definierte Typen (TYPE)
- Neu definierte Operatoren (OPERATOR, PRECEDENCE)
- Mehrstufige Strukturen, auch mit Bereichen als Komponenten
- Lokale Prozeduren
- Bolt-Variablen
- Variable Bit- und Zeichenketten-Ausschnitte

Komponenten des PEARL - Programmiersystems

Das portable PEARL-Programmierersystem von WERUM besitzt folgende Komponenten:

- Compiler (rechnerunabhängiges, einmal erstelltes Oberteil und rechnerabhängig zu erstellende Codegeneratoren)
- Laufzeitsystem, bestehend aus
 - Betriebssystem-Kern
 - Laufzeitpaket für binäre E/A
 - Laufzeitpaket für formatierte E/A
- Testsystem
- Datenbanksystem.

Rechnerabhängig wurden bei den bisherigen Implementierungen jeweils nur noch die Laufzeitroutinen für die PEARL-spezifische Arithmetik und Behandlung von Bit- und Zeichenketten, Prozeduren und Feldern etc. erstellt. Im übrigen werden die Standardkomponenten des Zielrechners (z.B. Binder, Editor) eingesetzt.

Im folgenden werden die Komponenten Compiler, Testsystem und Datenhaltungssystem näher vorgestellt.

PEARL - Compiler

Der PEARL-Compiler übersetzt PEARL-Programme je nach Rechner in Assembler oder verschiebbaren Bindercode. Die Handhabung der vielfältigen Compiler-Parameter ist oder wird der Bedienungsart angepaßt, die auf dem jeweiligen Zielrechner üblich ist.

Mittels solcher Compiler-Parameter können u.a. Listings der Quelle und des Übersetzungsergebnisses verlangt werden, wobei im Assembler- oder Bindercode-Listing Hinweise auf die Ursprungszeilen im Quellprogramm eingefügt sind. Außerdem erstellt der Compiler auf Wunsch eine Cross-Referenz-Liste aller benutzten Objekte mit Angaben der Quellzeilen-Nummern für ihre Definition und Benutzung. Durch Compiler-Parameter kann die Überwachung von Feldgrenzen und Referenzen zur Laufzeit ein- oder ausgeschaltet werden.

Der Compiler analysiert Programme umfangreich und recht genau. Die Fehlermeldungen erfolgen im Klartext mit Angabe der Quellzeilen-Nummer (423 verschiedene Fehlermeldungen sind möglich).

Ein Preprozessor gestattet das Einfügen von Programmtexten aus Files (%INCLUDE) und bedingte Compilierung (%IF).

Die Systemteil-Auswertung wird durch eine sogenannte Konfigurationsliste gesteuert, die alle Konfigurationsmöglichkeiten des Zielrechners beschreibt. Sollen diese Möglichkeiten erweitert werden, z.B. beim Anschluß eines bisher nicht vorgesehenen Gerätes, so kann der Anwender selbst mit einfachen Mitteln diese Konfigurationsliste anpassen und dem Compiler (sogar dynamisch) bekannt machen.

Auf Wunsch ist der Compiler in einer Version erhältlich, die bei entsprechender Einstellung eines Compiler-Parameters folgende Optimierungen durchführt:

- Nur einmalige Berechnung der Adressen von identischen Strukturkomponenten innerhalb einer PEARL-Anweisung.
- Nur einmalige Berechnung gemeinsamer Teilausdrücke.
- Direktes Absetzen fest indizierter Feldansprachen, d.h. keine Index-Berechnung zur Laufzeit in diesen Fällen.
- Rechnerabhängige Optimierungen.

Der Compiler ist in GBL1, einem echten PL/I-Subset, programmiert und kann somit einfach (als Cross-Compiler) auf Rechnern installiert werden, die über einen PL/I-Compiler verfügen (mit einem GBL1 umfassenden Sprachumfang; Beispiel: IBM).

Aufgrund der benutzten Compiler-Technologie kann der Compiler mit dem gleichen Code-generator auf Zielrechner transportiert werden, der für die Übersetzung von PEARL-Programmen erstellt wurde. Außerdem ist es möglich, den Compiler automatisch mittels eines existierenden Umsetzers auf FORTRAN-Rechner zu übertragen, um ihn dort als Cross-Compiler für andere Zielrechner oder als Compiler für diesen FORTRAN-Rechner einzusetzen.

Der PEARL-Compiler ist auf Rechnern folgenden Typs installiert (die Zahlen in Klammern geben den Ausbau der Rechner an):

- Amdahl 470/6
- Hewlett-Packard HP 1000 (64 KB)
- Hewlett-Packard HP 3000 (512 KB)
- Norsk Data NORD 10 S (128 KB)
- Siemens 310 (64 KB)
- Siemens 330 (128 KB)
- Siemens 7.760
- Siemens 404/3 (64 KB)

Auf den folgenden Rechnern wurden die erforderlichen Laufzeit-Komponenten installiert, so daß dort vom PEARL-Compiler übersetzte Programme ablauffähig sind:

- Hewlett-Packard HP 1000;
in Zusammenarbeit mit der TU Berlin; Termin: Ende 1980;
geplante Anwendungen: Steuerung von Experimenten und Prozessen im Prozeßrechnerverbund der TU Berlin.
- Hewlett-Packard HP 3000;
bisherige Anwendungen: U.a. Entwicklung rundfunkspezifischer Software im Institut für Rundfunktechnik (IRT), München.
- Norsk Data NORD 10 S / NORD 100;
in Zusammenarbeit mit Norsk Data; Termin: Herbst 1980;
geplante Anwendungen: U.a. Entwicklung von portabler Software für Analysen, Simulationen, Steuerungen im Halden Reactor Project, Norwegen.
- RDS (Really Distributed System: Mehrrechnersystem mit räumlich verteilten (Mikro-) Prozessorstationen mit Siemens 310-Befehlssatz);
in Zusammenarbeit mit dem Fraunhofer-Institut für Informations- und Datenverarbeitung (IITB), Karlsruhe;
bisherige Anwendungen: U.a. Regelung von Tiefofen bei der Thyssen AG, Duisburg.

PEARL - Compiler

Der PEARL-Compiler übersetzt PEARL-Programme je nach Rechner in Assembler oder verschiebbaren Bindercode. Die Handhabung der vielfältigen Compiler-Parameter ist oder wird der Bedienungsart angepaßt, die auf dem jeweiligen Zielrechner üblich ist.

Mittels solcher Compiler-Parameter können u.a. Listings der Quelle und des Übersetzungsergebnisses verlangt werden, wobei im Assembler- oder Bindercode-Listing Hinweise auf die Ursprungszeilen im Quellprogramm eingefügt sind. Außerdem erstellt der Compiler auf Wunsch eine Cross-Referenz-Liste aller benutzten Objekte mit Angaben der Quellzeilen-Nummern für ihre Definition und Benutzung. Durch Compiler-Parameter kann die Überwachung von Feldgrenzen und Referenzen zur Laufzeit ein- oder ausgeschaltet werden.

Der Compiler analysiert Programme umfangreich und recht genau. Die Fehlermeldungen erfolgen im Klartext mit Angabe der Quellzeilen-Nummer (423 verschiedene Fehlermeldungen sind möglich).

Ein Preprozessor gestattet das Einfügen von Programmtexten aus Files (%INCLUDE) und bedingte Compilierung (%IF).

Die Systemteil-Auswertung wird durch eine sogenannte Konfigurationsliste gesteuert, die alle Konfigurationsmöglichkeiten des Zielrechners beschreibt. Sollen diese Möglichkeiten erweitert werden, z.B. beim Anschluß eines bisher nicht vorgesehenen Gerätes, so kann der Anwender selbst mit einfachen Mitteln diese Konfigurationsliste anpassen und dem Compiler (sogar dynamisch) bekannt machen.

Auf Wunsch ist der Compiler in einer Version erhältlich, die bei entsprechender Einstellung eines Compiler-Parameters folgende Optimierungen durchführt:

- Nur einmalige Berechnung der Adressen von identischen Strukturkomponenten innerhalb einer PEARL-Anweisung.
- Nur einmalige Berechnung gemeinsamer Teilausdrücke.
- Direktes Absetzen fest indizierter Feldansprachen, d.h. keine Index-Berechnung zur Laufzeit in diesen Fällen.
- Rechnerabhängige Optimierungen.

Der Compiler ist in GBL1, einem echten PL/I-Subset, programmiert und kann somit einfach (als Cross-Compiler) auf Rechnern installiert werden, die über einen PL/I-Compiler verfügen (mit einem GBL1 umfassenden Sprachumfang; Beispiel: IBM).

Diese Ausbaustufe läßt sich auf größeren Zielrechnern und auf Cross-Rechnern installieren. Bei der Cross-Version werden das PEARL-Betriebssystem des Zielrechners (unter Verwendung des PEARL-Betriebssystemkerns von WERUM) und die Zeitachse auf dem Cross-Rechner insoweit simuliert, daß das Zusammenspiel der Tasks in zeitlich richtiger Reihenfolge erfolgen kann.

Bis Ende 1980 soll diese Ausbaustufe auf HP 3000 und Siemens 330 verfügbar sein.

Stufe 3:

Die dritte Ausbaustufe wird in der Regel auf einem Cross-Rechner installiert. Sie enthält zusätzlich zur zweiten Ausbaustufe die folgenden Testhilfen:

- Operationsgenaue Simulation des Zeitverhaltens von Zielrechnern,
- Simulation der E/A von Zielrechnern mittels
 - E/A im Dialog mit dem Bediener,
 - Gegentasks,
 - Files mit Testdaten,
- Breakpoints auch auf
 - zeitliche Ereignisse (analog PEARL-Schedules),
 - Interrupts,
 - E/A-Anweisungen,
- Deadlock-Analyse und
- Interrupt-Anweisungen.

Die Implementation dieser dritten Ausbaustufe ist in Vorbereitung für die Rechner HP 3000 und Siemens 330.

PEARL - Datenbanksystem

Zur Unterstützung der PEARL-Programmierung von Automationssystemen mit großem Datenhaltungsanteil entwickelt WERUM ein Echtzeit-Datenbanksystem für den simultanen Zugriff durch PEARL-Programme und Dialog-Benutzer auf gemeinsame Datenbestände. Wesentliche Merkmale dieses Systems sind:

- Strategiefreie Ansprache,
d.h. die Ansprache der Daten durch Dialogkommandos oder Anweisungen im PEARL-Programm ist unabhängig von der software-technischen Realisierung der Datenorganisation und der Zugriffsverfahren.

- Offenheit,
d.h. die implementierten, speziellen Zugriffsverfahren können einfach ausgetauscht werden ohne die Ansprache der Daten in den PEARL-Programmen oder Dialog-Kommandos zu ändern.

Durch diese beiden Eigenschaften kann das Datenbanksystem bei der Automation von Prozessen flexibel eingesetzt oder mit relativ geringem Aufwand zeitlich parallel zur Erstellung der Anwendungssoftware speziellen Anforderungen angepaßt werden.

Im Dialogbetrieb werden die folgenden Funktionen mittels der Dialogsprache DIALOG verfügbar sein:

DIALOG-Verwaltungsanweisungen ("DDL-Kommandos"):

- Datenbanken und Dateien einrichten und entfernen,
- definieren und modifizieren,
- Zugriffsverfahren definieren, modifizieren, (Dateien) zuordnen und entfernen,
- Dateiverknüpfungen definieren, modifizieren und aufheben.

DIALOG - Abfrage- und Änderungsanweisungen ("QL-Kommandos"):

- Sätze einer Datei auch unter komplexen Nebenbedingungen und Bezügen zu anderen Dateien anwählen, am Bildschirm zeigen, auf dem Drucker ausgeben und/oder in benutzer-eigene Dateien übernehmen,
- Listen von Sätzen erstellen,
- Sätze in Dateien einfügen, ändern oder löschen,
- Sätze einer Datei Sätzen einer verknüpften Datei zuordnen,
- Zuordnungen zwischen Sätzen aufheben.

Zur Ansprache der Daten in PEARL-Programmen werden die Funktionen in Form von logischen Mengen-Definitionen und Mengen-Anweisungen zur Verfügung stehen, die durch eine Vor-Übersetzung auf äquivalente PEARL-Prozeduraufrufe abgebildet werden.

Ziel ist, dieselbe Datenhaltungsschnittstelle wie in DIPOL (Discrete Process Oriented Language) zu realisieren.

Die erste Ausbaustufe dieses Datenbanksystems ist auf Siemens 330 realisiert worden: Sie umfaßt die oben aufgeführten Funktionen mit Ausnahme der Funktionen für Dateiverknüpfungen; die Ansprache der Daten in PEARL-Programmen muß noch mit Prozeduraufrufen erfolgen. Außerdem folgt noch kein automatischer Wiederanlauf nach Fehlern oder Ausfall.

Die erste Ausbaustufe bietet neben sequentiellen Zugriffsverfahren ein modifiziertes B-Baum-Verfahren für den direkten Zugriff; derzeit werden ein Hash-Verfahren und ein Bitlisten-Verfahren zugefügt.

Die laufenden Erweiterungsarbeiten umfassen zudem weitere Maßnahmen zur Datensicherung.

Das Datenbanksystem ist/wird portabel in PEARL und GBL1 programmiert und kann somit auf denselben Rechnern wie der PEARL-Compiler installiert werden.

Seine Entwicklung erfolgt mit Mitteln des BMFT im Rahmen eines Forschungsvorhabens des Projekts Fertigungstechnik (PFT) des Kernforschungszentrums Karlsruhe GmbH.