



PEARL-News

Ausgabe 1 Januar 2014

Mitteilungen
des GI/GMA/ITG-Fachausschusses
Echtzeitsysteme

ISSN 1437-5966

Impressum

Herausgeber	GI/GMA/ITG-Fachausschuss Echtzeitsysteme URL: http://www.real-time.de
Sprecher	Prof. Dr. Dr. W. A. Halang FernUniversität in Hagen Fakultät für Mathematik und Informatik Lehrstuhl für Informationstechnik, insb. Realzeitsysteme D-58084 Hagen Telefon: 02331/987-372 E-Mail: wolfgang.halang@fernuni-hagen.de
Stellvertreter	Prof. Dr. D. Zöbel Universität Koblenz-Landau Fachbereich Informatik Institut für Softwaretechnik Postfach 201 602 D-56016 Koblenz Telefon: 0261/287-2724 E-Mail: zoebel@uni-koblenz.de
Redaktion	Prof. Dr.-Ing. habil. H. Unger FernUniversität in Hagen Fakultät für Mathematik und Informatik Lehrstuhl für Kommunikationsnetze D-58084 Hagen Telefon: 02331/987-1155 E-Mail: herwig.unger@fernuni-hagen.de
ISSN	1437-5966

Redaktionell abgeschlossen am 24. Januar 2014

Einreichung von Beiträgen: Beiträge zu dieser Zeitschrift sind jederzeit hoch willkommen. Je früher und „rechtzeitiger“ sie eintreffen und je inhaltlich interessanter und hochwertiger sie sind, umso glücklicher ist der Redakteur. Einreichungen „auf den letzten Drücker“ und das Textverarbeitungsprogramm Word verleiten ihn hingegen zu Wutausbrüchen. Darum bittet er höflich um Übersendung der Beiträge per elektronischer Post in reinem ASCII und nicht codiert – eben ohne jeden Firlefanz – oder noch besser in LaTeX.

Inhalt

- 1 Tagungsankündigung „Echtzeit 2014“
- 2 Ehrenmitgliedschaft für Prof. Dr. Leberecht Frevert
- 3 Graduiertenwettbewerb 2013
- 4 Best Paper Award 2013
- 5 Vorwort der Tagung „Echtzeit 2013“
- 6 Workshop „Autonome Systeme“ in Cala Millor, Mallorca
- 7 Betrachtung der Typen CHAR und REF CHAR in PEARL90
- 8 Kreativität ist (weiterhin) gefragt

1 Tagungsankündigung „Echtzeit 2014“

In diesem Jahr findet die Fachtagung „Echtzeit“ am 20. und 21. November 2014 in Boppard am Rhein statt. Das Leitthema wird lauten: „**Industrie 4.0 und Echtzeit**“. Der Einsendeschluss für Vortragsanmeldungen ist Sonntag, der 20. April 2014. Nähere Informationen finden Sie unter <http://www.real-time.de/workshop.html>.

2 Ehrenmitgliedschaft für Prof. Dr. Leberecht Frevert

Der Fachausschuss hat Prof. Dr. Leberecht Frevert in Anerkennung seiner Leistungen auf dem Gebiet der Echtzeitsysteme und der Entwicklung der Programmiersprache PEARL die Ehrenmitgliedschaft verliehen. Leider war es Prof. Frevert nicht möglich, die Urkunde persönlich entgegen zu nehmen.

3 Graduiertenwettbewerb 2013

Das Programmkomitee prämierte folgende Abschlussarbeiten als herausragend:

Alexander Beifuß (Universität Hamburg) Leistungs- und Präzisionssteigerung des Lastgenerierungsprozesses von UniLoG unter Verwendung echtzeitfördernder Maßnahmen durch das Betriebssystem

Rainer Müller (Friedrich-Alexander-Universität Erlangen-Nürnberg) Slothful Linux: An Efficient Hybrid Real-Time System by Hardware-Based Task Dispatching

Manuel Strobel (Hochschule Furtwangen) Entwurf und Implementierung einer Prozessinterkommunikation für Multi-Core CPUs



von links: Halang, Beifuß, Strobel, Müller, Benra

Jutta Düring (Tagungsorganisation)

4 Best Paper Award 2013

Das Programmkomitee hat den Beitrag „IT-Sicherheitseigenschaften für eng gekoppelte, asynchrone Multi-Betriebssysteme im automotiven Umfeld“ von Pierre Schnarz und Joachim Wietzke (Hochschule Darmstadt) als besten Tagungsbeitrag prämiert.



von links: Halang, Schnarz, Zöbel

Jutta Düring (Tagungsorganisation)

5 Vorwort der Tagung „Echtzeit 2013“

Programmierbare elektronische Systeme werden in einer Fülle sicherheitsgerichteter Anwendungen eingesetzt. Sie übernehmen Aufgaben zur Überwachung oder Steuerung medizinischer Geräte, chemischer Anlagen, von Anti-Blockier-Systemen, Luft- oder Weltraumfahrzeugen, Fertigungsmaschinen sowie in Kraftwerken und der Energieverteilung. Weil falsch erstellte Systeme oder schlicht Fehler darin zum Versagen der Systemfunktionen führen können, was schwere Schäden verursachen oder gar Menschenleben gefährden kann, müssen solche eingebetteten Systeme hohe Sicherheitsanforderungen erfüllen. Der industrielle Bedarf an sicherheitsgerichteten, programmgesteuerten Systemen ist hoch und steigt durch die zunehmende Automatisierung von Prozessen kontinuierlich weiter an. Im Einklang damit wächst auch das gesellschaftliche Sicherheitsbewusstsein.

Dies sind die Gründe, warum die Fachtagung Echtzeit in diesem Jahr das Leitthema funktionale Sicherheit aufgreift, aber auch, weshalb der GI/GMA/ ITG-Fachausschuss Echtzeitsysteme damit begonnen hat, die Echtzeitprogrammiersprache PEARL so weiterzuentwickeln, dass sowohl die funktionale Sicherheit in ihr geschriebener Programme erhöht als auch der Zustand erreicht werden, dass sich rechnergestützte, ggf. verteilte Systeme mit einem Grad an Vertrauen in ihre Verlässlichkeit erstellen lassen, der ihre Zulassung für sicherheitskritische Steuer- und Regelaufgaben durch die Aufsichtsbehörden auf der Basis formeller Abnahmen erlaubt. Diese Weiterentwicklung soll bis zur Formulierung einer neuen DIN-Norm vorangetrieben werden, um die bisherigen PEARL-Normen DIN 66253-2 und DIN 66253 Teil 3 abzulösen. Die zukünftige Norm wird jeweils geeignete, inhärent sichere Sprachteilmengen zur Erstellung von Anwendungen definieren, die den Sicherheitsintegritätsstufen SIL 1 bis SIL 4 nach IEC 61508 genügen müssen. So wird die weltweit einzige, im Hinblick auf funktionale Sicherheit konzipierte Programmiersprache entstehen. Aber davon wird an anderer Stelle zu berichten sein. Möglicherweise wird der Tagungsband des nächsten Jahres die ersten Ergebnisse enthalten.

Normen sind für die funktionale Sicherheit von besonderer Bedeutung. Es könnte wohl keinen Geeigneteren als den Referenten des eingeladenen Vortrages geben, der innerhalb der zuständigen Organisation, der Deutschen Kommission Elektrotechnik Elektronik Informationstechnik, seit rund zwei Jahrzehnten die einschlägigen Gemeinschaftskomitees „Grundsätze für Rechner in Systemen mit Sicherheitsaufgaben“ und „Funktionale Sicherheit elektrischer, elektronischer und programmierbarer elektronischer Systeme zum Schutz von Personen und Umwelt“ betreut, Einblick in die Denkweise der Sicherheitstechnik sowie die Normungsarbeit zu geben. Wegen ihrer inhärenten Fehleranfälligkeit muss insbesondere sicherheitsgerichtete Software auf Normenkonformität hin geprüft werden. Um dieses schwierige und zeitraubende Unterfangen zu systematisieren und bei der Prüfung nichts auszulassen, bietet sich deren Unterstützung mit Werkzeugen an, so wie sie beispielhaft im zweiten Beitrag vorgestellt wird.

Die Problematik der funktionalen Sicherheit programmierbarer elektronischer Systeme tritt seit einigen Jahren in großem Umfang in der Automobilindustrie auf, weil dort zunehmend Funktionen, die früher mechanisch, elektromechanisch oder hydraulisch realisiert wurden oder die es früher noch gar nicht gab, durch eingebettete Systeme implementiert werden. Als Themen aus diesem Bereich werden deshalb in der zweiten Sitzung ein optisches Sensorsystem für die Einknickwinkel zwischen Zugfahrzeug und Zweiachsanhänger ebenso behandelt wie die Anforderungen an Betriebssysteme, die gemeinschaftlich in komplexen, am Fahrer angebrachten Multi-mediasystemen arbeiten sollen.

Seit Langem liegt dem Fachausschuss die Förderung des Nachwuchses besonders am Herzen, weshalb er Studierende aufruft, ihre Abschlussarbeiten zu einem jährlichen Graduiertenwettbewerb einzureichen. Die Sieger erhalten nicht nur Preise, sondern auch Gelegenheit, sich und ihre Arbeiten auf der Tagung zu präsentieren. Die drei prämierten Arbeiten dieses Jahres beschäftigen sich mit dem Einfluss von Betriebssystemeigenschaften auf die Qualität von Lastgeneratoren, hybriden Betriebssystemen zur Verringerung von Echtzeitanforderungen sowie Interprozesskommunikation auf Mehrkernrechnern.

Die vierte Sitzung ist der Entwicklung sicherer Systeme gewidmet. Nach einem Konzept zum Aufbau fehlertoleranter verteilter Echtzeitsysteme aus standardisierten, mit mehreren Schnittstellen zu verschiedenen Bussystemen ausgestatteten Einplatinenrechnern werden Verfahren zur empirischen Bestimmung von Programmausführungszeiten auf Mehrkernprozessoren sowie zur statistischen Synthese von Modellparametern zur Sicherheitsanalyse hybrider Systeme vorgestellt.

Vor ihrem Einsatz gilt es, sicherheitsgerichtete Systeme zu verifizieren und zu validieren. Damit beschäftigt sich die abschließende Sitzung. Im Rahmen der Integration vernetzter elektronischer Systeme müssen die Aktivitäten der einzelnen Teilnehmer simuliert werden. Sicherheitsgerichtete Anwendungen speicherprogrammierbarer Steuerungen aus verifizierten Bibliotheken entnommenen Funktionsblöcken zusammensetzen, ist ein aufwandsreduzierender Ansatz. Die resultierenden Funktionspläne werden zur Verifikation in formale Modelle transformiert. Schließlich werden programmgesteuerte Systeme zur qualitativen Analyse ihrer funktionalen Sicherheit durch eine Kombination von Zustandsdiagrammen und fehlerbaumtypischen Gattern modelliert: erstere für die zeitlichen Beziehungen und letztere für die kausalen Zusammenhänge.

Den Autoren sei gedankt, ihre Beiträge meistens pünktlich und in vorgegebener Form abgeliefert zu haben. Auf die redaktionelle Feinarbeit an diesem Band sowie Fehlerkorrektur hat Frau Dipl.-Ing. Jutta Düring wieder viel Mühe verwendet, wofür ich ihr besonders herzlich danken möchte. Für die auch in diesem Jahr gewährte finanzielle Unterstützung der Fachtagung in Boppard ist der Fachausschuss den langjährigen industriellen Unterstützern Ingenieurbüro für Technik Cseke und Domeyer GbR und Beenen IT-Lösungen zu großem Dank verpflichtet.

6 Workshop „Autonome Systeme“ in Cala Millor, Mallorca

Wie bereits seit 6 Jahren trafen sich auch im Oktober 2013 wieder 20 – 30 Wissenschaftler auf der Insel Mallorca, um Fortschritte und Herausforderungen auf dem Gebiet der autonomen Systeme zu besprechen. Dieses Jahr war es jedoch das erste Mal, dass der Arbeitskreis Echtzeitkommunikation des Fachausschusses Echtzeitsysteme offizieller Mitveranstalter war. Der Grund liegt in der Wandlung des Workshops von einem Doktorandentreffen mit potentiellen Gutachtern in eine Plattform für Wissenschaftler und Forscher, um neue Ideen auszuarbeiten und sich von Lösungen anderer Forschungsgebiete inspirieren zu lassen. Hierbei profitiert der Workshop sowohl von der familiären Atmosphäre als auch durch seine Abgeschlossenheit in einem Hotel weit weg vom täglichen Geschäft.

Die Beiträge des Tagungsbandes 2013 geben einen Überblick über die neuen Forschungstätigkeiten und Interessen der Teilnehmer im Zusammenhang mit autonomen Systemen, im Detail:

- Schnelle **Hardware** und – heute mehr denn je – hardwarebasierte Sicherheitslösungen bilden die Grundlage autonomer Systeme. Die Reihe der präsentierten Arbeiten reicht von Netzwerklösungen on-chip bis zu erforderlichen Prüfverfahren.
- **Algorithmen** sind die Basis verschiedener Softwarelösungen. Die meisten Probleme entstehen dadurch, korrekte Klassifikationen zu bilden, Eigenschaften und Beschaffenheit eines Systems zu erkennen und hieraus die richtigen Empfehlungen abzuleiten. Die Sicherheitsaspekte der Kommunikation in verteilten Systemen wurden hierbei nicht vernachlässigt.
- Immer wenn autonome Einheiten, Agenten oder Systeme in einer gemeinsam genutzten Umgebung agieren, tritt eine direkte oder indirekte Kommunikation und/oder Zusammenarbeit auf, und in den meisten Fällen findet **Selbstorganisation** statt. Das dritte Kapitel befasst sich mit solchen Strukturen, ihrem Erscheinen, ihrer Analyse und Einschätzung.



Nach dem Erfolg der Workshopreihe wird es am gleichen Ort vom 26.–30. Oktober 2014 ein weiteres Treffen geben. Interessierte Mitglieder aus dem Fachausschuss werden gebeten, sich spätestens bis zum 15.3.2014 bei Herwig Unger (herwig.unger@fernuni-hagen.de) zu melden, da wie bislang die Veranstaltung auf Einladungsbasis und mit beschränkter Teilnehmerzahl stattfinden wird. Nähere Informationen findet man unter <http://www.fernuni-hagen.de/kn/seminar14/>.

Herwig Unger, FernUniversität in Hagen

7 Betrachtung der Typen CHAR und REF CHAR in PEARL90

7.1 Einleitung

Im Rahmen der Realisierung einer Laufzeitumgebung für SmallPEARL [4] bin ich auf eine Problematik mit den Typen CHAR und REF CHAR gestoßen. Als Basisinformation dient hier der DIN-Standard [1] sowie das weitgehend identische Werk PEARL90-Sprachreport [2]. Alle für diesen Artikel relevanten Abschnitte sind in [1] und [2] inhaltlich identisch. Eine weitere Quelle ist das RTOS-UH-Handbuch [3].

Ein Grundprinzip bei der Sprachdefinition von PEARL ist die Berechenbarkeit des Programms. D.h. es sind keine Sprachkonstrukte enthalten, die erst zur Laufzeit problematisch werden können. Daher gibt es z.B. keine dynamische Speicherverwaltung und nur feste Tasks.

7.2 Begriffsklärung

Der Datentyp CHAR beschreibt in PEARL90 einen Datentyp mit einer festen Länge. Es ist auch festgelegt, dass bei Zuweisungen einer kürzeren Zeichenkette auf eine längere CHAR-Variable diese rechts mit Leerzeichen aufgefüllt wird. Was im umgekehrten Fall passieren soll ist leider nicht festgelegt.

Beispiel:

```
DCL x CHAR(10); /* Zeichenkette mit 10 Zeichen */

x := 'Hallo'; /* ergibt die Zeichenkette 'Hallo      ' */
```

Der Typ REF CHAR beschreibt einen Datentyp, der eine flexible Verwaltung über eine CHAR-Variable legt, so dass der tatsächliche Füllstand bekannt ist und nicht automatisch die Restlänge mit Leerzeichen aufgefüllt wird. Falls der Zielstring kürzer als der Quellstring ist, so sollte dies in der Sprachnorm geregelt werden. Eine Compilerfehlermeldung erscheint angebracht, da die Typen nicht kompatibel sind.

```
DCL rc REF CHAR();
DCL content CHAR(100);
rc := content; // rc has maximum length and actual length of 100

CONT rc := 'HALLO'; // actual length = 5; max length = 100
```

7.2.1 Operator CAT

Der Operator CAT verkettet zwei Character-Variablen in der Weise, dass ein Ergebnis vom Typ der Gesamtlänge der beteiligten Operanden erzeugt wird. Nach Sprachreferenz dürfen bei CAT normale CHAR-Variablen wie auch REF CHAR-Variablen verwendet werden.

Für die CAT-Operation ist die Größe des Ergebnisses vom Compiler berechenbar, sofern alle beteiligten Quelloperanden vom Typ CHAR sind. Sobald mindestens ein REF CHAR-Operand beteiligt ist, kann der Compiler diese Ergebnisgröße nicht mehr berechnen.

Bei einer Zuweisung führt dies zu keinem Problem, da sich die maximale Größe des Ergebnisses aus dem Zielloperanden ableiten lässt. Diese Größe ist bei normalen CHAR-Variablen dem Compiler bekannt, bei REF CHAR ist dies dem Laufzeitsystem bekannt. Eine beliebige Verkettung lässt sich dann von links nach rechts auflösen. Der Zielloperand wird dazu zunächst – falls er noch kein

REF CHAR-Typ ist – in einen derartigen Typ übergeführt und dessen aktuelle Länge auf 0 gesetzt. Dann werden die einzelnen Operanden von links nach rechts an das Ziel angehängt. Falls es dabei zu einer Längenüberschreitung kommt, so wird das Laufzeitsystem ein PEARL-Signal auslösen.

7.2.2 Operator .CHAR(a,b)

Mit dem Operator .CHAR(a,b) kann ein Teil einer CHAR-Variable herausgeschnitten werden (Selektion). Das Ergebnis ist eine CHAR-Variable mit der Länge $b - a + 1$. Diese Länge ist vom Compiler berechenbar, wenn

1. a und b konstant sind, oder
2. $b - a$ konstant ist.

Für diese Fälle kann der Compiler eine entsprechende Ergebnisvariable anlegen.

Falls die Werte a oder b für den Compiler nicht berechenbare Werte einnehmen können, so muss das Ergebnis von Typ REF CHAR sein. Die maximale Größe dieses Ergebnisses ist für den Compiler von der Größe des Quelloperanden ableitbar. Damit ist dies übersetzbar, aber nicht ideal. Im Fall einer Zuweisung ist dieser Schritt nicht notwendig, da das gleiche Verfahren wie in Abschnitt 7.2.1 benutzt werden kann.

7.3 CHAR als Prozedurparameter

Wenn CHAR-Variablen an Prozeduren übergeben werden, so kann der formale Prozedurparameter unterschiedlich definiert werden. Je nach Definitionsart kann der Compiler unterschiedlich agieren.

feste CHAR-Länge Der Compiler kennt die Größe des formalen Parameters und kann ein temporäres Objekt erzeugen und eine Kopie des aktuellen Parameters zuweisen. Es gibt keine Probleme bei Ausdrücken mit CAT und REF CHAR.

REF CHAR() Der Compiler übergibt ein REF CHAR-Objekt. Dieses wird per Referenz übergeben. Es wird insbesondere keine Kopie des aktuellen Parameters angelegt.

- Sofern der aktuelle Parameter eine bekannte Länge hat, ist dies unproblematisch.
- Falls der aktuelle Parameter ein REF CHAR-Objekt ist, so ist dies auch einfach möglich.
- Falls der aktuelle Parameter ein Ausdruck mit REF CHAR ist, so kann der Compiler die Größe des Ergebnisses nicht ermitteln und damit auch keinen sicheren Code erzeugen. Dies muss daher zu einem Übersetzungsfehler führen. Ausdrücke mit CAT sind dann genauso verboten wie .CHAR(a,b)-Selektionen mit REF CHAR-Ergebnis.

7.4 CHAR als Datenquelle und Datensenke bei CONVERT

Die CONVERT-Anweisung erlaubt es, die Formatierungsanweisung auf Character-Strings anzuwenden anstelle eine DATION zu benutzen. Hier ist explizit vorgesehen, dass Character-Variablen mit fester Länge zum Einsatz kommen. Für die interne Verarbeitung ist es für das Laufzeitsystem von Vorteil, eine REF CHAR-Variable über die Datensenke zu legen, um einfach die einzelnen Formatelemente anzuhängen.

7.5 Auswirkungen auf den Übersetzungsvorgang

Bei Zuweisungen von Konkatenationen werden die einzelnen Teilstrings nacheinander an den Zielstring über ein REF CHAR-Objekt zugewiesen. Als letzte Operation wird ggf. ein CHAR(x)-Objekt aus dem REF CHAR () gebaut. Damit treten bei Zuweisungen keine Probleme mit vielen REF CHAR-Typen und .CHAR(a,b)-Selektionen auf.

Beispiel:

```
DCL (a,b,c,d) FIXED;
DCL x CHAR(10);
DCL y CHAR(100);
DCL rc REF CHAR();
rc := x;

/* do some calculations on a,b,c,d,x,y */
....
CONT rc := 'hallo' CAT x.CHAR(a,b) CAT y.CHAR(c,d);
```

wird im Pseudocode umgesetzt als:

```
....
SIZEOF rc LENGTH := 0;
APPEND(rc,'hallo');
APPEND(rc,x.CHAR(a,b));
APPEND(rc,y.CHAR(c,d));
```

Bei Erstellung von aktuellen Prozedurparametern wird im Fall eines festen CHAR-Typs ein entsprechendes temporäres Objekt erzeugt und der aktuelle Parameter (ggf. auch mit einem beliebig gearteten CHAR-Ausdruck) darauf zugewiesen.

Falls der formale Prozedurparameter ein REF CHAR()-Typ ist, so muss ein beliebiger .CHAR(a,b)-Ausdruck zu einem Übersetzungsfehler führen, da der Compiler keinen Platz für den aktuellen Parameter sinnvoll berechnen kann¹. Als pragmatische Lösung wird vorgeschlagen, in diesem Fall nur feste CHAR-Stringausdrücke oder einfache REF CHAR()-Variable als aktuelle Parameter zuzulassen. Dieser muss vom Anwendungsprogramm vorab berechnet werden.

7.6 Fazit

Die Behandlung von REF CHAR ist in der Norm von PEARL90 nicht eindeutig geregelt. In der Sprachreferenz ist auch noch die Ungereimtheit, dass die Länge einer CHAR-Variable vom Typ Fixed(15) ist, REF CHAR() haben als Typ für die Längeninformation allerdings FIXED(31).

Beim Vergleichsalgorithmus wird festgelegt, dass die kürzere Variable vor dem Vergleich mit Leerzeichen aufgefüllt wird. Dies führt im nachstehenden Beispiel zu einem unerwartetem Ergebnis:

```
DCL x CHAR(10) INIT('HALLO');
DCL y CHAR(11);

y := x CAT TOCHAR(0);
```

¹Die Lösung, die Summe aller maximalen Längen zu bilden, führt zu einem sehr hohen Speicherverbrauch.

```
IF (x LT y) THEN
    /* will never reached */
ELSE
    ...
FIN;
```

Bezüglich RTOS-UH [3] ist noch vermerkt, dass hier keine Selektionen und REF CHAR unterstützt werden.

Bei der Erstellung des Laufzeitsystems für SmallPEARL [4] wird das hier skizzierte Verhalten realisiert, welches sehr sparsam mit den Rechnerressourcen umgeht. Ebenso werden die Typen für die Längen von CHAR und REF CHAR einheitlich als FIXED(31) angesetzt. SmallPEARL steht unter BSD-Lizenz.

Rainer Müller, Hochschule Furtwangen

Literatur

- [1] DIN; DIN66253-2 Programmiersprache PEARL – PEARL90; Beuth Verlag; Berlin; 2012
- [2] GI; PEARL90-Sprachreport; 1995
- [3] W. Gerth; RTOS-UH; 2006
- [4] M. Schaible, R. Müller; SmallPEARL; <https://sourceforge.net/p/smallpearl>

8 Kreativität ist (weiterhin) gefragt

Die Suche nach einem neuen Namen für unsere „PEARL-News“ ist noch nicht beendet. Der bisher beste Vorschlag lautet:

Die *Echt*Zeit

Wer einen besseren Vorschlag hat, möge ihn senden an tagung@real-time.de. Einzige Voraussetzung ist die Verbindung zum Begriff 'Echtzeit'.

Jutta Düring, FernUniversität in Hagen