Die versteckten Fallen der "Jahr-2000"-Fähigkeit von Software am Beispiel von REXX

Zusammenfassung: Um Software "Jahr-2000"-fähig zu machen, müssen Programmierer auch auf Fallen achten, die sich durch die interne Datenrepräsentation ergeben können. Am Beispiel der Sprache REXX wird dies mit einer entsprechenden Lösungsmöglichkeit aufgezeigt

Autorin/Autor: Dr. Erwin Hoffmann, fehcom Köln (Deutschland)
erwin_hoffmann@csi.com

Ausgabe: 1.0

Datum: 30.09.1999

Referenz: GE9001 (www.mannherz.de)

Die Inhalte dieser Seiten dürfen ohne vorherige schriftliche Zustimmung des Herausgebers nicht verändert und nicht öffentlich vorgeführt oder für kommerzielle Zwecke vervielfältigt werden. Dies gilt auch für eine Nutzung der Inhalte dieser Seiten auf anderen Web-Seiten oder vernetzten Rechnern.

Diese Seiten und die enthaltenen Informationen wurden nach bestem Wissen und Gewissen zusammengestellt, eine Rechtsverbindlichkeit kann hieraus nicht abgeleitet werden. Die Haftung für die Richtigkeit, Vollständigkeit und Verlässlichkeit von Informationen, Daten und Dokumenten wird auf die Fälle grob fahrlässigen oder vorsätzlichen Verhaltens beschränkt für jegliche Verluste oder Schäden (auch Folgeschäden), die dadurch enstehen, dass die Nutzerin oder der Nutzer auf Informationen vertraut, die er im Rahmen der Nutzung des Dienstes erhalten hat. Ebenfalls wird die Haftung auf die Fälle grob fahrlässigen oder vorsätzlichen Verhaltens beschränkt für jegliche Verluste oder Schäden (auch Folgeschäden), die der Nutzerin oder dem Nutzer auf andere Weise bei der Nutzung dieser Seiten entstehen (z. B. durch Herunterladen von Web-Seiten o. ä.).

Soweit wir vertraglich gegenüber bestimmten Kunden dazu verpflichtet sind, bestimmte Inhalte auf unseren Web-Seiten zur Verfügung zu halten, wird die Haftung auf die Fälle grob fahrlässigen oder vorsätzlichen Verhaltens beschränkt für jegliche Verluste oder Schäden (auch Folgeschäden), wobei dies für die leicht fahrlässige Verletzung vertragswesentlicher Pflichten nicht gilt.

Verantwortlich für den Inhalt ist die Autorin bzw. der Autor.

Inhaltsverzeichnis

1. Das Jahr 2000: Die versteckten Fallen von REXX

1.1 Problemstellung

Das neue Jahrtausend vor Augen sind Software-Entwickler intensiv dabei, vorhandene Programme "Jahr-2000"-fähig zu machen. Grundsätzlich müssen dabei die folgenden drei Probleme gelöst werden:

1. Alle EDV-Programme sollten das Jahr als Zahl aus vier Ziffern darstellen.
2. Das Jahr 2000 sollte als Jahr "2000" und nicht als "00" oder "1900" interpretiert werden.
3. Das Jahr 2000 muss korrekt als Schaltjahr identifiziert werden, dies stellt - dem Gregorianischen Kalender folgend - eine Ausnahme des normalen Vierjahreszyklus dar.

Fazit ist, dass jede Berechnung, die Datums- und Zeitvergleiche zwischen einem beliebigen Datum im Jahr 2000 und anderen Jahren ermittelt, das korrekte Ergebnis liefern muss.

Es sei den Philosophen überlassen, einen "guten" Grund dafür herauszufinden, warum EDV-Programme oft das Jahr mit lediglich zwei Ziffern darstellen, selbst am Ende des ausgehenden Jahrhunderts. Während es recht einfach ist, dem Betriebssystem und dem BIOS eine konsistente (vierstellige) Ausgabe der Jahreszahl "beizubringen" und eine korrekte Zählweise für das Jahr 2000 sicherzustellen, können individuelle Programme trotzdem unvorhersehbare Resultate erzeugen - an dieser Stelle entsteht weiterer Aufwand. Bei diesen Programmen ist es notwendig, jeden Datums- und Zeitaufruf, der falsche Werte erzeugen kann, zu ermitteln und unverzüglich zu korrigieren.

1.2 Datenrepräsentation

Im täglichen EDV-Leben werden Datum und Zeit nicht als Variablensatz wie "`day=01`", "`month=August`" oder "`month=08`" and "`year=1999`" vorliegen, sondern - um Speicherplatz zu sparen - in einer einzelnen Variable abgelegt, etwa im Format "`date=01081999`" oder "`date=19990801`", abhängig vom Programmierer. Natürlich ist es ebenso naheliegend, das Datum und die Zeit in einer Variable kombiniert abzulegen, beispielsweise als "`time=199908011110`". In diesem Beispiel stellt der erste Teil der Variable "time" das Datum als "1999/08/01" und der zweite Teil die Zeit als "11:10" dar. Vorteilhaft an dieser Notation ist dabei, dass die Zeitachse vollständig enthalten ist, d. h. ein Vergleich zweier Datums-/Zeitwerte läuft auf eine einfache Subtraktion der entsprechend formatierten Variablen hinaus. Aber auch hier gilt, dass Daten im "alten" Format, etwa "9807030411", erkannt und vor der Weiterverarbeitung in das neue Format, also "199807030411" umgewandelt werden müssen, was glücklicherweise einfach zu bewerkstelligen ist.

1.3 Programmiersprache REXX

REXX - eine Programmiersprache, die in IBM® Umgebungen (OS/2®, MVS™, AIX®) weit verbreitet ist - hat eine Besonderheit, die in diesem Zusammenhang etwas ausführlicher erklärt werden muss. Zeitmarken wie "`time=19990801110301`" können als Variable eines speziellen Typs definiert werden. Wie bekannt, können Variablen in den Programmiersprachen als Fließkommazahl, Ganzzahl, Zeichenfolge oder in anderen Formaten - je nach Sprache - deklariert werden.

Zur arithmetischen Weiterverarbeitung würde jeder Programmierer also eine Variable vom Typ Ganzzahl (Integer) heranziehen. Eine Ausnahme davon bilden Werte, die auch Bruchteile der Sekunden angeben sollen, etwa "`time=19990801110301.0112`", aber diese Verwendung ist eher unwahrscheinlich. Deshalb scheint die Benutzung des Typs Ganzzahl (Integer) völlig ungefährlich zu sein. Aber Vorsicht! Jeder Wert im Ganzzahlformat oder im Fließkommaformat ist normalerweise in der Größe (bzw. in der Stellenanzahl) beschränkt. Oft resultiert diese Beschränkung aus der Abbildung der Variablen auf ein einziges Computerwort, bei den meisten Betriebssystemen besteht dieses aus vier Byte, also 32 Bit. Der größte darstellbare Wert ist demnach 2^{32}-1 - groß genug, möchte man denken. Der zugehörige Ganzzahlwert ist dann 2^{32}-1=4294967295 und damit kleiner als jeder der genannten Zeitmarken!

Standardmäßig verwendet REXX eine Genauigkeit von neun Ziffern unabhängig vom gewählten Format. Werden die großen ganzzahligen Zeitmarken in Berechnungen verwendet, meldet REXX keinerlei Fehler - die Werte werden einfach an dieser Obergrenze abgeschnitten. Im Fall von Vergleichen (innerhalb eines IF-THEN-ELSE Konstrukts, zum Beispiel) führt dieses Abschneiden jedoch zu unvorhersehbaren Ergebnissen.

© 1999 Mannherz EDV-Dienstleistungen (www.mannherz.de)

Diesem Effekt kann allerdings relativ leicht begegnet werden. REXX beachtet die Anweisung NUMERIC DIGITS, die globale Gültigkeit hat und die die Länge (Anzahl der Ziffern) jeder Variable erweitert auf den Wert N, wenn in der aufrufenden Hauptroutine des Programms "`NUMERIC DIGITS N`" angegeben wurde.

1.4 Beispielprogramm

Die folgende REXX-Funktion zeigt die Verwendung der Anweisung NUMERIC DIGITS am Beispiel eines Programms, das unter OS/2® Warp 3 die einfache Zeitmarke einer Datei liefert. Das Programm kann eigenständig oder als Teil eines weiteren REXX-Programms benutzt werden:

```
/* REXX-Function FileDate */
/* Erwin Hoffmann (5.9.1997) - Year 2000 proof */
/* FileDate: */

NUMERIC DIGITS 15

arg file

datefile = ""
date0=9000000000

if file = "" then signal kein
if file = "?" | file = "/?" then signal hilfe

/* OS/2 Libraries */
call RxFuncAdd 'SysLoadFuncs', 'RexxUtil', 'SysLoadFuncs'
call SysLoadFuncs

/* Algorithm */
rc=sysfiletree(file,datum,"bt")
if rc > 0 | datum.1 = "DATUM.1" then signal nicht
if datum.0 > 1 then signal mehr
datefile=word(datum.1,1)
datefile=space(translate(datefile," /","/ "),0)
if datefile > date0 then datefile = 19||datefile
 else datefile = 20||datefile
/* Return */
/* return(datefile) */
say datefile
exit

nicht: /* File not found */
say "<FileDate> File not found."
say datefile
exit
/* return(datefile) */

mehr: /* More then one File given */
say "<FileDate> is suited for one single File only."
say datefile
exit
/* return(datefile) */

kein:  /* No Input-File given */
say "<FileDate> No INPUT File given."
exit

hilfe: /* Give Help */
say "<FileDate> Return Value: 'YEAR-MONTH-DAY-HOUR-MINUTE'"
exit
```

© 1999 Mannherz EDV-Dienstleistungen (www.mannherz.de)

2. Marken

- Das Logo ist eine in Deutschland eingetragene Marke von Mannherz EDV-Dienstleistungen
- AIX ist eine eingetragene Marke von IBM Corp. in den Vereinigten Staaten von Amerika und/oder anderen Ländern
- IBM ist eine eingetragene Marke von IBM Corp. in den Vereinigten Staaten von Amerika und/oder anderen Ländern
- MVS ist eine Marke von IBM Corp.
- OS/2 ist eine eingetragene Marke von IBM Corp. in den Vereinigten Staaten von Amerika und/oder anderen Ländern

Alle anderen auf diesen Seiten erwähnten Produkt- bzw. Firmennamen sind Marken ihrer jeweiligen Eigentümer.

© 1999 Mannherz EDV-Dienstleistungen (www.mannherz.de)