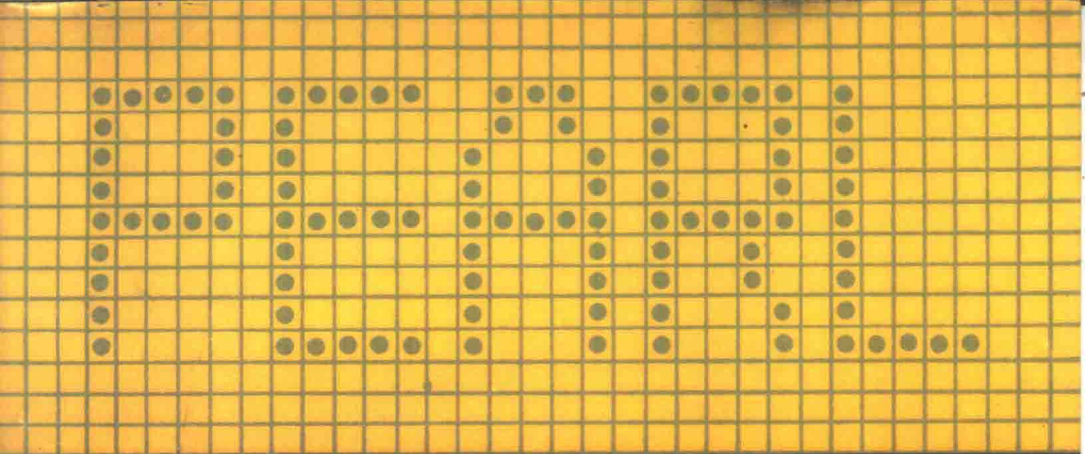




过程与实验自动化实时语言

●〔西德〕W.维鲁姆 H.温道尔 著

● 中国标准出版社

过程与实验自动化实时语言 (PEARL)

〔西德〕 W.维鲁姆 H.温道尔 著

惠敦炎 汪健飞 译
周师道 单立群

中国标准出版社

Wulf Werum/Hans Windauer

PEARL

Process and Experiment
Automation Realtime Language

Vieweg

(1978)

过程与实验自动化实时语言
(PEARL)

〔西德〕 W.维鲁姆 H.温道尔 著
惠敦炎 汪健飞 周师道 单立群 译

•
中国标准出版社出版
(北京复外三里河)

中国标准出版社秦皇岛印刷厂印刷
新华书店北京发行所发行 各地新华书店经售

•
开本 850×1168 1/32 印张 5³/₈ 字数 141,000

1986年2月第一版 1986年2月第一次印刷

印数 1—5,000

•
书号: 15169·3-295 定价 1.85 元

•
标目 31—3

内 容 提 要

PEARL 是一种典型的用于过程控制的高级实时语言。目前国际上流行的许多过程控制计算机都配有 PEARL 语言的编译程序。在西德，这种语言已列为 DIN 标准，在许多领域获得日益广泛的应用。本书详细阐述了 PEARL 的主要特征、基本功能以及其他功能。书末附录包括关键词一览表、句法索引等，便于读者查阅。

本书可供从事计算机发展和应用以及过程自动化方面的工程技术和科研人员阅读，亦可供大专院校有关专业师生参考。

译者序

PEARL 是一种典型的、用于过程控制的高级实时语言。它是西德的一些计算机制造厂、软件公司、研究中心和大学在联邦研究与技术部(BMFT)的促进之下,从1969年至1976年共同研制而成的。1978年,

“Basic PEARL”就已经列为西德 DIN 标准 (DIN 662531, I)。1980 年, “Full PEARL” 也列为 DIN 标准 (DIN 66253, I)。

迄今为止,已有 30 多种中型和微小型计算机配有这种语言的编译程序。在西德, PEARL 日益广泛地应用于金属加工、电力、化工、仓库管理、电视、采矿、运输、宇航、教育、气象预报等领域。据报导,同 ASSEMBLER 语言相比, PEARL 的应用不仅在程序设计方面,而且在以后的系统维护方面,至少可以节省 50% 的时间和费用。PEARL 对任何硬件环境的适应能力、实时响应能力、并行处理能力以及模拟过程现场的能力,远远胜过其他过程控制语言。它做为开发过程控制应用软件的工具,是很有发展前途的,应予以足够的重视。

本书是根据西德 W. 维鲁姆(W. Werum)和 H. 温道尔 (H. Windauer) 合著的《PEARL-Process and Experiment Automation Realtime Language》一书 1978 年德文版翻译的。原著示例中的德文词或德文注释均按照 1982 年英文版(书名为“Introduction to PEARL, Vieweg, 1982”)改为英文。原著与 1982 年英文版在个别细节上有所差异,译本中凡引用 1982 年英文版的地方均加了译者注。书末附录 3.4 句法索引表由按英文字母顺序排列改为按汉语拼音字母顺序排列。

西德 PEARL 协会主席、斯图加特大学教授 R. 劳伯 (Prof. R. Lauber, Universitaet, Stuttgart), 《PEARL 评论》杂志 (PEARL-Rundschau) 副总编辑、卡尔斯卢厄核研究中心 T. 马丁博

士 (Dr.T.Martin, Kernforschungszentrum, Karlsruhe) 和本书作者维鲁姆软件开发公司 H.温道尔博士 (Dr.H.Windauer, Entwicklungsbuero, Wulf Werum, Lueneburg) 对于本书的翻译出版十分关注并给予了热情的支持。在审定过程中, 张诚、邵鸿声等同志提出了宝贵的意见。在此谨向他们一并表示衷心的感谢。

由于时间仓促, 译者水平有限, 书中误漏之处在所难免, 恳请读者批评指正。

1983 年 8 月

原 序

过程与实验自动化实时语言 PEARL (Process and Experiment Automation Realtime Language 的缩写) 是一种高级程序设计语言, 对于应用电子计算机控制生产过程的一切领域中的实时任务, 可以在相当大的程度上与具体计算机无关地进行程序设计。

本书定义了 PEARL 子集的内容, 多特蒙德数学咨询和程序设计服务公司 (mbp) 及吕内堡 W. 维鲁姆软件开发公司 (Entwicklungsbuero, Wulf Werum, Lueneburg) 已把该子集在流行的过程计算机上实现, 它由编译和操作系统所组成。

该子集是 “Full PEARL” 的一部分, 但已包括了全部的 “Basic PEARL”。

“Basic PEARL” 已为 AEG, BBC, Dietz, GPP, Krupp-Atlas, mbp, 西门子和维鲁姆等公司所采纳, 这是 PEARL 已实现的最低限度的规模。mbp-PEARL 除了包括 Basic PEARL 以外, 还包括了 Full PEARL 的一些语言元, 从而可以为任务调度和高级操作系统, 特别是表格处理方面进行巧妙而有效的程序设计。对 “Full PEARL” 的缩减主要涉及到在 PEARL 程序运行时对内存和管理要求额外开销的那些语言元。本书附录中对这些语言元作了确切的描述。

本书引论中概略地描述了 PEARL 的主要特征, 这些正是与其他众所周知的高级编程语言如 FORTRAN 或 PL/I 的主要区别。为了便于读者更好地理解, 在引论后的第二篇中, 仅介绍基本功能, 这些对于编写简单的 PEARL 程序已经够用了。其他功能则放在本书第三篇中讨论。

附录包括关键词一览表、数据类型及其适用范围、句法 (由于教学上的原因不可能在各章节中详尽地叙述所有的句法问题) 以及对

“Full PEARL”所作缩减的功能的说明。

在编写本书时，作者参考了以下著作，尤其是有关“Basic PEARL”的著作，从中获益不浅。

- Basic PEARL Language Description. Gesellschaft fuer Kernforschung mbH, Karlsruhe;
PDV-Report KFK-PDV 120, 1977.
- Full PEARL Language Description. Gesellschaft fuer Kernforschung mbH, Karlsruhe;
PDV-Report KFK-PDV 130 (近期出版)。
- Die Arbeitspapiere des PEARL-Subset-Arbeitskreises.
- mbp-PEARL, Sprachreport, mbp und Werum, 1975.

对于 R.布鲁门他尔 (R. Blumenthal), T.吕斯特博士 (Dr. T. Roestel), G.R.沙勃尔 (Gerd R. Sapper) 和 H.-G.伏云克 (H.-G. Vowinkel) 先生们的审议和指正，作者在此谨表谢意。

目 录

第一篇 引论.....	(1)
1. PEARL 的主要特征.....	(1)
1.1 实时特性.....	(1)
1.2 输入和输出.....	(2)
1.3 程序结构.....	(3)
2. PEARL 语言形式的构成规则.....	(3)
2.1 字符集.....	(4)
2.2 基本元素.....	(5)
2.2.1 标识符.....	(6)
2.2.2 数值常量.....	(6)
2.2.3 串常量.....	(7)
2.2.4 时间常量.....	(8)
2.2.5 注释.....	(8)
2.3 语言形式的构成.....	(9)
第二篇 基本功能.....	(11)
1. 程序结构.....	(11)
2. 问题数据.....	(13)
2.1 标量型问题数据	(14)
2.1.1 整型变量	(15)
2.1.2 浮点型变量	(15)
2.1.3 位串变量	(15)
2.1.4 字符串变量	(16)
2.1.5 时钟变量	(16)
2.1.6 间隔时间变量	(16)
2.2 问题数组	(17)
3. 过程 (Procedures)	(18)
3.1 过程说明	(20)

3.2 过程调用.....	(22)
4. 并行活动.....	(25)
4.1 任务 (Task) 说明.....	(26)
4.2 中断	(27)
4.3 任务控制语句	(28)
4.3.1 启动条件.....	(28)
4.3.2 任务的启动.....	(31)
4.3.3 任务的结束.....	(33)
4.3.4 任务的挂起.....	(33)
4.3.5 任务的继续.....	(33)
4.3.6 任务的延迟.....	(34)
4.3.7 任务的撤消.....	(35)
4.4 多重任务同步.....	(36)
4.4.1 借助于 Sema (信号灯) 变量的独占访问和同步.....	(37)
4.4.2 依靠 Bolt (闩) 变量对资源分配的控制.....	(41)
5. 表达式, 赋值.....	(43)
5.1 表达式	(43)
5.1.1 单操作数运算符.....	(46)
5.1.2 双操作数运算符.....	(47)
5.1.3 表达式的运算.....	(53)
5.2 赋值.....	(54)
6. 程序运行顺序控制语句.....	(55)
6.1 转移语句.....	(56)
6.2 条件语句.....	(57)
6.3 选择语句和空语句.....	(58)
6.4 循环语句.....	(60)
7. 输入、输出.....	(62)
7.1 系统部分.....	(63)
7.2 问题部分数据站的约定.....	(67)
7.3 数据站的打开和关闭.....	(72)
7.4 读 (Read) 语句和写 (Write) 语句.....	(75)
7.5 取 (Get) 语句和存 (Put) 语句.....	(81)
7.5.1 定点格式.....	(85)

7.5.2 浮点格式.....	(86)
7.5.3 字符串格式.....	(88)
7.5.4 二进制位格式.....	(89)
7.5.5 时钟格式.....	(90)
7.5.6 间隔时间格式.....	(90)
7.5.7 表格式.....	(91)
7.5.8 重复格式.....	(92)
7.6 收 (Take) 语句和发 (Send) 语句.....	(93)
第三篇 其他功能.....	(94)
1. 结构.....	(94)
2. 位串和字符串的使用.....	(98)
3. 新数据类型的约定.....	(101)
4. 用参照变量间接寻址.....	(102)
5. 对象的程序块结构,有效域.....	(105)
6. 模块之间的联系.....	(108)
7. 初始化属性.....	(111)
8. 赋值保护.....	(112)
9. 驻留属性.....	(113)
10. 重进入属性.....	(113)
11. 运算符.....	(114)
11.1 用于类型变换的运算符.....	(114)
11.2 其他的标准运算符.....	(115)
11.3 新运算符的约定.....	(117)
12. 中断语句.....	(119)
13. 信号.....	(122)
14. 其他的输入输出功能.....	(124)
15. 长度约定.....	(126)
附 录.....	(128)
1. 附缩写的关键词一览表.....	(128)
2. 数据类型及其适用范围.....	(132)
3. 句法.....	(133)
3.1 基本元素,程序.....	(133)
3.2 问题部分.....	(135)

3.2.1 说明.....	(136)
3.2.2 规定.....	(141)
3.2.3 语句.....	(142)
3.3 系统部分.....	(148)
3.4 句法索引表.....	(148)
4. 相对 Full-PEARL 所作 缩 减.....	(157)

第一篇 引 论

1. PEARL的主要特征

1.1 实时特性

用来对一个技术过程 (process) 进行在线控制或分析的数据处理程序应能对该过程发出的信息或有时间性的事件尽快地作出反应。单一的顺序程序, 即按固定的时间顺序来安排和执行的程序不能满足上述要求。于是, 不如把复杂程度不一的自动化任务按问题的性质分解为紧迫性各不相同的分任务, 并使程序的结构适应于任务的结构。由此而产生了服务于分任务的这样一些独立的程序段: 它们可以在时间顺序上安插在其他任务 (例如: procedure “过程”) 中来处理。但也产生了服务于分任务的一些独立的程序单元, 它们受时间上并非确定的事件的激励 (例如来自技术过程的干扰), 而必须立即同一切其他任务并行地加以处理。执行这样一个程序单元称为 Task (任务)。各种 Task 可通过预先规定其优先级而确定其紧迫程度。

PEARL 语言为多个 Task 的约定和多个 Task 技术过程之间的协调提供了以下各种可能性:

• Task 约定, 如

```
SUPPLY, TASK PRIORITY 2;  
任务体 (约定, 语句)  
END;
```

• 启始 (激活), 如

```
ACTIVATE SUPPLY;
```

• 结束, 如

```
TERMINATE PRINTING;
```

• 挂起, 如

```
SUSPEND STATISTICS;
```

· 继续, 如

CONTINUE STATISTICS;

· 延迟, 如

AFTER 5 SEC RESUME;

为适应自动化任务的需要, 某些语句能安排重复执行, 如钟点到达某时间的情况、经过某一时间间隔或出现某一信息的情况:

WHEN READY ACTIVATE SUPPLY;

(含义: 每当出现“READY”中断时立即启动任务“SUPPLY”。)

当然, 也可以预先安排时间上周期地启动:

AT 12:00 EVERY 30 MIN UNTIL 15:00
ACTIVATE PROTOCOL;

各个 Task 只要不受到相应措施的限制便各自执行其语句。然而, 有时候也要求对两个或多个 Task 实现同步。例如, 某个 Task 为另一个 Task 产生数据并存入缓冲器。此时就不允许数据产生者工作得比数据使用者更快。此外, 还有更复杂的同步问题。例如, 某个 Task 为写要独占地访问某文件, 而其他 Task 为读可同时访问该文件。为解决这类同步问题, PEARL 中设置了信号灯 (Semaphore) 变量和闩 (Bolt) 变量。

1.2 输入和输出

借助于同具体计算机无关的程序语句可以实现计算机标准外围设备 (打印机、穿孔卡片阅读机、磁盘等) 或过程外围设备 (检测仪表、执行机构等) 之间的数据传送以及文件管理。

外围设备和文件统称为数据站。

主要有两种不同的数据传送方式:

· 无格式控制 (即以计算机内部格式表示) 的数据传送: 这种数据传送方式适用于允许顺序存取和直接存取的文件传送, 以及适用于过程数据传送。

例如:

READ ARTICLE FROM ARTICLEFILE BY

POS (1φ) ;
SEND OFF TO MOTOR (I) ;

· 有格式控制的数据传送，即应用于计算机内部格式同数据站上规定的外部表示之间的转换：

用数据站字符集的字符来描述数据（字符型格式化类似于 PL/I 语言）。

例如：

PUT RESULT TO PRINTER BY F (5) ;

数据站的名称可以自由选择。这是通过将 一个 PEARL 程序分成与计算机有关的和基本上无关的两大部分来达到的。

1.3 程序结构

为解决综合自动化问题，程序系统应当是模块化的。PEARL 是通过这种方式来满足这一要求的，即 一个 PEARL 程序是由一个或几个可分别编译的模块所组成。为能与具体计算机无关地编写数据传输语句，以及安排对技术过程信息（中断）或计算机硬件信号的反应，通常将一个模块分成“系统”部分和“问题”两部分。

“系统”部分描述硬件的配置。特别应当指出的是，这里对系统各设备及其连接方式，中断和信号均可用自由选择的名称来命名。请看下例：一个仓库货架搬运车 RFZ 的控制器连接于数字输出装置 DIGOUT (3) 的第 2 至 9 号接线端；

RFZ : DIGOUT (3) * (2 : 9) ; — >

这里的 DIGOUT (3) 是数字输出装置在计算机系统中的专有名称；而接线端 2 至 9 则得到自由选择的与具体计算机无关的用户名称 RFZ。

“问题”部分利用上述在“系统”部分中引入的用户名称就能编写出与具体计算机无关的程序来，以解决自动化的任务，例如：

WRITE OUTPUT TO RFZ ;

2. PEARL 语言形式的构成规则

PEARL 程序为自由格式，即不用专门的程序格式书写，尤其是

不必考虑一个语句应从那一列开始写。

所有的 PEARL 的程序元均由下列字符集里的字符所组成。此外，字符串和注释则可以包含机器配置所认可的任意一个字符。

2.1 字符集

PEARL 的字符集包括下列元素：

- 大写字母 A~Z
- 数字 0~9
- 专用符号
 - ┐ 传送符（空格、换行、卡片末尾、回车等），
 - ' 引号，
 - (左圆括号，
 -) 右圆括号，
 - , 逗号，
 - . 句号，
 - ; 分号，
 - : 冒号，
 - + 加号（如：用于加法，前置符号），
 - 减号（如：用于减法，前置符号），
 - * 星号（如：用于乘法），
 - / 斜杠（如：用于除法），
 - = 等号（如：用于赋值），
 - < 小于号，
 - > 大于号，
 - ⌊ 左结构方括号，
 - ⌋ 右结构方括号

符号⌊和⌋在程序中通常都以方括号“[”和“]”的形式出现。这里采用下加横道的形式是为了与语言形式描述中出现的表示可选的括号相区别。

下列符号组合只能作为一个整体（组合符号）来使用：

- • 指数符，

/* 注释开始,
 */ 注释结束,
 // 整除符,
 == 相等符,
 /= 不等符,
 <= 小于等于符,
 >= 大于等于符,
 <> 带 C 循环移位符,
 >< 串接符。

如果编写程序的设备不具备上述所有符号, 则可用下列字符来代替:

LT 代替 <
 GT 代替 >
 NE 代替 /=
 LE 代替 <=
 GE 代替 >=
 CSHIFT 代替 <>
 CAT 代替 ><
 (/ 代替 ≡
 /) 代替 ≡

2.2 基本元素

PEARL 程序由下列基本元素构成:

- 标识符,
- 数值常数,
- 串常量,
- 时间常量,
- 分隔符 (特殊字符及组合符号),
- 注释。

在标识符、数值常量、时间常量和串常量之后必须跟着分隔符或注释。