

Lecture Notes in Computer Science

Edited by G. Goos and J. Hartmanis

198

Arnold Businger

PORTAL Language Description

Second, Extended Edition



Springer-Verlag

Berlin Heidelberg New York London Paris Tokyo

Editorial Board

D. Barstow W. Brauer P. Brinch Hansen D. Gries D. Luckham
C. Moler A. Pnueli G. Seegmüller J. Stoer N. Wirth

Author

Arnold Businger
LGZ Landis & Gyr Zug Corp.
CH-6301 Zug, Switzerland

CR Subject Classification (1987): D.1.3, D.2.1–2, D.3.1, D.3.3

ISBN 3-540-18960-2 Springer-Verlag Berlin Heidelberg New York
ISBN 0-387-18960-2 Springer-Verlag New York Berlin Heidelberg

ISBN 3-540-15682-8 1. Aufl. Springer-Verlag Berlin Heidelberg New York
ISBN 0-387-15682-8 1st ed. Springer-Verlag New York Heidelberg Berlin

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in other ways, and storage in data banks. Duplication of this publication or parts thereof is only permitted under the provisions of the German Copyright Law of September 9, 1965, in its version of June 24, 1985, and a copyright fee must always be paid. Violations fall under the prosecution act of the German Copyright Law.

© by Landis & Gyr 1985, 1988
Printed in Germany

Printing and binding: Druckhaus Beltz, Hemsbach/Bergstr.
2145/3140-543210

PREFACE

The aim of this book is to give a systematic and detailed description of the programming language PORTAL, a high level language for the programming of reliable real-time systems, developed by LGZ Landis & Gyr Zug Corp. in Zug, Switzerland. In contrast to the introductory text "Programming in PORTAL" this language definition is more in the style of a manual. As a rule, the reader is expected to be familiar with a high level programming language such as Pascal, or to have acquired some PORTAL knowledge either in a course or through self-study. If these conditions are fulfilled, the language description may be used as a text-book or as a reference manual.

In the first chapter R. Schild explains the goals of PORTAL. In addition this introductory chapter contains a description of the Backus-Naur form -- the meta-language used to describe the PORTAL syntax in the remaining chapters.

The second chapter describes the lexical symbols of the language. Here the reader will find answers to questions such as "what characters may be used in a name?" or "how is a comment delimited?"

Chapters three to seven describe constants, data types, variables, expressions, and statements. Most of these language elements appear in the same or similar form in Pascal. But there are some constructions that are specific to PORTAL: the index type (4.1.7), the case type (4.4), the string type (4.6), the loop-statement (7.4.3), the with-statement (7.5), and the using-statement (7.6).

Chapters eight through ten treat routines, processes, and modules. These program blocks differ from the corresponding Pascal constructions (functions and procedures) or are not even available (resources, processes and modules). The concepts described in these chapters are of primary importance; they are therefore explained in detail.

Chapter eleven shows the overall structure of a PORTAL program. The reader will learn how to construct a complete program, and also what the influence of the block structure is on the scope of the identifiers.

Appendix A contains information on the different implementations of PORTAL. This consists of technical data (e.g. the range of the type INTEGER, or the implementation of sets) as well as certain deviations from the language definition (restrictions and additions).

Appendix B describes the machine-dependent elements of PORTAL. These include constructs such as AT, which serves to position variables at programmer-defined machine addresses.

Appendix C describes a number of compiler control commands.

Appendix D contains a summary of all the standard routines available in PORTAL.

The bibliography at the end of the book refers the reader to other publications on PORTAL.

Finally, I want to thank all those who have helped and supported me in this work and whose critical remarks and suggestions have improved this

book. In particular I would like to mention P. Bucherer, J. Hunt, W. Joerg, P. Kopp, H. Lienhard, M. Meier, M. Meyer, E. Neuenschwander, H. Oswald, B. Steinle, O. Thaler, and J. Zingg.

Special thanks go to R. Schild and R. Schoenberger. Without their continued readiness to answer all sorts of questions this language definition would never have come into existence.

Zug, July 1983

A. Businger

Preface to the 1st English Edition

The original German version has been translated into English without any major changes. Some minor errors have been corrected, and two new features in the language -- constants computed at compile time, and the possibility to add assembler code -- have been included. To give the reader a feeling for the language in actual use, a new Appendix E presents a complete PORTAL program, consisting of a module to control an elevator, together with a simulation environment.

I would like to thank my wife Carol for doing such an excellent job of translating; many thanks also go to numerous colleagues for their help with the additions and corrections.

Zug, May 1985

R. Schild

Preface to the 2nd edition

To our delight PORTAL continues to be successful. In order to improve the usefulness of this book as a reference, we have added Appendices F and G with a collection of the BNF definitions of PORTAL. In addition a minor correction was made on page 34.

Zug, January 1988

R. Schild

TABLE OF CONTENTS

1. Introduction	1
1.1. The goals of PORTAL	1
1.2. Syntax representation	2
2. Lexical symbols	4
2.1. The alphabet	4
2.2. Basic definitions	4
2.3. The structure of program texts	5
2.4. Comments	6
2.5. Options	6
2.6. Identifiers	6
2.7. Reserved words	7
2.8. Numeric literals	8
2.8.1. Unsigned integers	8
2.8.2. Unsigned reals	8
2.8.3. Unsigned numbers	9
2.9. Character Literals	9
2.10. Character Strings	10
3. Constants	11
3.1. Definition and examples	11
3.2. The constant declaration	12
3.2.1. The simple constant declaration	13
3.2.2. The construction of string constants	14
4. Data types	15
4.1. Simple data types	15
4.1.1. The standard type INTEGER	16
4.1.2. The standard type BOOLEAN	17
4.1.3. The standard type CHAR	18
4.1.4. The standard type REAL	19
4.1.5. Symbolic types	20
4.1.6. Subrange types	21
4.1.7. Index types	22
4.1.8. Operations and standard routines	24
4.2. Array types	25
Array operations	26
4.3. Record types	27
Record operations	28
a) Selective access to components	29
b) The "with" notation	29
4.4. Case types	30
Case type operations	31
a) Access to the tag field	31
b) The case-statement	32
c) The with-set-statement	32
d) Access when the value of the tag field is known ...	33
4.5. Set types	34
Set operations	34
a) Set constructors	34
b) Set relations	35
c) Set operations	36

4.6.	String types	37
	String manipulations	38
	a) String constructors	38
	b) String selectors	39
	c) Assignments	40
	d) String relations	41
	e) Standard functions	42
4.7.	The type declaration	43
5.	Variables	45
5.1.	Definition and examples	45
5.2.	Introducing variables	47
	5.2.1. The variable declaration	47
	Compatibility	47
	5.2.2. Temporary variables	48
	5.2.3. Read-only variables	49
6.	Expressions	51
6.1.	Definition and examples	51
6.2.	The evaluation of expressions and operator precedence.....	53
	The evaluation of boolean expressions	54
6.3.	The function call	55
7.	Statements	59
7.1.	The assignment-statement	59
	a) The ordinary assignment	59
	b) Special assignments	60
	c) Type compatibility	60
7.2.	The procedure call	61
7.3.	Selective statements	65
	7.3.1. The if-statement	65
	7.3.2. The case-statement	67
	a) The ordinary case-statement	67
	b) The case-statement for case types	68
7.4.	Iterative statements	70
	7.4.1. The while-statement	71
	7.4.2. The repeat-statement	71
	7.4.3. The loop-statement	72
7.5.	The with-statement	73
	a) The with-statement for convenient access to the (exported) routines of a module	73
	b) The with-statement for variables	74
	c) The with-set-statement	75
	d) The with-statement for expressions	76
	e) Temporary variables	76
	f) Nesting of with-statements	77
7.6.	The using-statement	78
8.	Routines	82
8.1.	The declaration of a routine	82
	a) The routine header	83
	b) The routine body	84
8.2.	Functions	85
	a) The function header	85
	b) The function body	86
	c) Examples	87
8.3.	Procedures	87
	a) The procedure header	88
	b) The procedure body	88

c) Examples	89
8.4. Resources	90
a) The resource header	91
b) The resource body	91
c) Examples	92
8.5. Routine types	94
8.6. Recursion	96
The standard resource RECURSION	97
9. Processes and their synchronization	100
9.1. Parallel processes	100
9.1.1. Sequential and parallel activities	100
9.1.2. The activities of a process	101
9.1.3. The interaction of processes	102
9.1.4. The advantages of parallel processes	103
9.2. Process declarations	103
9.3. The synchronization of parallel processes	104
9.3.1. Monitors	104
9.3.2. Signals	106
The signal declaration	106
The WAIT-statement with delay	107
The standard function AWAITED	108
9.4. An example	108
10. Modules	112
10.1. The module concept	112
10.1.1. The characteristics of modules	112
Interaction with the environment	112
Initialization	113
10.1.2. Monitors	115
10.2. The module declaration	115
10.3. The use of modules	116
a) The module as a program package	117
b) The module as a data structure	120
c) Exporting data types from a module	120
10.4. Module types	121
10.4.1. The declaration of module types	121
10.4.2. The module instantiation	123
10.4.3. The use of module types	124
a) Multisort	124
b) Stack	125
11. The structure of a program	128
11.1. The basic structure	128
11.2. The block structure and the scope of names	128
11.2.1. Local objects	129
Temporary names	130
Name conflicts	131
11.2.2. The import of objects	131
11.2.3. The export of objects	133
"Defines" out of a resource routine	135
11.3. The lifetime of blocks	135
Appendix A. Remarks on the implementation	137
A.1. Technical data	137
A.1.1. The character set	137
A.1.2. The data type INTEGER	137
A.1.3. The data type LONGINT	137
A.1.4. The data type BOOLEAN	138

A.1.5.	The data type CHAR	138
A.1.6.	The data type REAL	139
A.1.7.	The symbolic types	139
A.1.8.	The subrange types	140
A.1.9.	The set types	140
A.1.10.	The arrangement of variables	141
A.1.11.	The WAIT-statement with delay	142
A.1.12.	Code size limits	142
A.2.	Restrictions	143
A.2.1.	The construction of sets	143
A.2.2.	The format of case-statements	143
A.2.3.	The call of a routine with only one parameter	143
A.2.4.	The data type of a function result	143
A.2.5.	The call of GRANT	144
A.2.6.	Recursion	144
A.2.7.	The initialization code of modules	144
A.2.8.	The structure of module types	145
A.2.9.	Multi-level "defines" of data types	145
A.2.10.	Calling of "defined" routines	145
A.3.	Computed constants: restrictions	145
Appendix B.	Machine dependent language elements	147
B.1.	Declarations with "AT"	147
B.1.1.	The positioning of record components	147
B.1.2.	The placement of variables	148
B.2.	The standard functions "ADDR" and "ADDRHIGH"	148
B.3.	"ADDRESS" and "GRANTADDR"	149
B.4.	The standard function "SIZE"	150
B.5.	External signals	151
B.6.	The specification of priorities for processes and monitors	151
Rules for the use of priorities	152	
B.7.	The standard procedure "FATAL"	152
B.8.	The standard procedure "RSX"	153
Appendix C.	Options	154
a)	Controlling the program listing	154
b)	Inclusion of assembler code	155
c)	Options for run-time tests	156
Appendix D.	Summary of available standard routines	157
a)	Standard functions	157
b)	Standard procedures	158
c)	Standard resource	159
Appendix E.	A sample program	160
Appendix F.	List of all BNF definitions	183
Appendix G.	Alphabetical list of the BNF definitions	190
Bibliography	194
Index	195