

Obsah

	Předmluva	11
	ČÁST I	
1	Požadavky na návrh RT-jazyků	13
1.1	Charakteristika RT-systémů	13
1.2	Základní vlastnosti RT-jazyků	16
1.2.1	Zabezpečení	16
1.2.2	Čitelnost	17
1.2.3	Flexibilita	18
1.2.4	Jednoduchost	18
1.2.5	Portabilita	19
1.2.6	Efektivita	19
1.3	Požadavky na RT-jazyky	19
2	Jednoduché datové typy	21
2.1	Základní pojmy	21
2.2	Mechanismy typové kontroly dat	23
2.2.1	Slabá typová kontrola dat	23
2.2.2	Silná typová kontrola dat	25
2.2.3	Odvozené typy	26
2.2.4	Ekvivalence typů	27
2.2.5	Dědičnost atributů	28
2.2.6	Zúžení	30
2.2.7	Podtypy	31
2.2.8	Anonymní typy a podtypy	32
2.3	Výčetové typy	33
2.3.1	Výčetové typy definované uživatelem	34
2.3.2	Boolovské typy	36
2.3.3	Znakové typy	37
2.4	Numerické typy	39
2.4.1	Celočíselné typy	40
2.4.2	Typy s pohyblivou řádovou čárkou	42
2.4.3	Typy s pevnou řádovou čárkou	44
2.5	Koncepce typů v současných programovacích jazycích	52
3.	Strukturované datové typy	54
3.1	Strukturování dat	54
3.2	Pole	55

3.2.1	Základní vlastnosti	55
3.2.2	Typový mechanismus polí	56
3.2.3	Vicerozměrná pole	58
3.2.4	Konstruktory polí	58
3.2.5	Implementace polí	59
3.3	Záznamy	59
3.3.1	Pevné záznamy	59
3.3.2	Variantní záznamy	61
3.3.3	Sjednocení	63
3.3.4	Varianty, nebo sjednocení	64
3.3.5	Implementace záznamů	65
3.4	Množiny	65
3.4.1	Typ množina	65
3.4.2	Implementace množin	67
3.5	Dynamické datové struktury	67
3.5.1	Statické a dynamické proměnné	67
3.5.2	Problémy návrhu	71
3.5.2.1	Typová kontrola dynamických dat	72
3.5.2.2	Dereference	74
3.5.2.3	Vytváření a rušení dynamických datových objektů	75
3.5.3	Problémy implementace	77
3.6	Strukturované datové typy v konkrétních programovacích jazycích	79
4	Klasické programové struktury	81
4.1	Struktura programu	81
4.2	Řídící příkazy	82
4.2.1	Základní druhy řídicích příkazů	82
4.2.2	Návrh řídicích příkazů	83
4.2.3	Příklady návrhu základních řídicích příkazů	85
4.2.4	Předčasné ukončení příkazů cyklu	87
4.3	Bloková struktura	88
4.3.1	Motivace	88
4.3.2	Bloky	89
4.3.3	Implementace bloků	92
4.4	Procedury	92
4.4.1	Podstata procedur	92
4.4.2	Parametry	94
4.4.3	Pravidla platnosti jmen	99
4.4.4	Problémy implementace	101
4.5	Funkce a operátory	102
4.5.1	Funkce	102
4.5.2	Přetěžování	104
4.5.3	Operátory	107
4.6	Programové struktury v konkrétních programovacích jazycích	109

5	Mechanismy abstrakce	111
5.1	Abstraktní datové typy	111
5.2	Širší souvislosti zavedení modulů	114
5.3	Návrh mechanismu modulu	115
5.3.1	Obecná struktura modulu	115
5.3.2	Specifikace rozhraní	117
5.3.3	Pravidla viditelnosti jmen	119
5.4	Implementace	121
5.5	Mechanismus abstrakce v konkrétních programovacích jazycích	122
6	Paralelismus	124
6.1	Paralelismus v RT-systémech	124
6.2	Tradiční přístup k paralelnímu zpracování	125
6.3	Procesy	130
6.4	Monitory	132
6.5	Rendezvous	136
6.6	Předávání zpráv	141
6.7	Paralelní zpracování v RT-úlohách	142
6.8	Problémy implementace	145
6.9	Paralelismus v konkrétních programovacích jazycích	147
7	Přímé programování periferních operací	149
7.1	Nástin problematiky	149
7.2	Přístup k technickým prostředkům	151
7.2.1	Adresování registrů periferií	152
7.2.2	Reprezentace registrů pomocí uživatelských typů	153
7.2.3	Periferní moduly	155
7.3	Přímé programování periferních operací	156
7.3.1	Mechanismy technického vybavení	156
7.3.2	Abstraktní model	157
7.3.3	Jazykový model	158
7.4	Programování vstupu/výstupu	161
7.4.1	Přenos znaků	161
7.4.2	Přenos bloků užitím DMA	164
7.5	Přímé programování periferních operací v konkrétních programovacích jazycích	167
8	Ošetření chyb	168
8.1	Základní pojmy	168
8.2	Klasické metody ošetření chyb	169
8.3	Obsluhy výjimek	172
8.3.1	Strukturované ošetření chyb	172
8.3.2	Základní mechanismy obsluhy výjimek	175
8.3.3	Šíření výjimek	178
8.3.4	Zotavení z chyb	180
8.3.5	Obsluhy výjimek v paralelních procesech	183

8.3.6	Problémy implementace	185
8.4	Ošetření chyb v konkrétních programovacích jazycích	187
9	Návrh jazyka – shrnutí	189
9.1	Typová kontrola dat	189
9.2	Struktura programu	192
9.3	Paralelismus	193
9.4	Přímé programování vstupu/výstupu	195
9.5	Ošetření chyb	197
9.6	Další problémy návrhu programovacího jazyka	198
9.6.1	Separátní kompilace	198
9.6.2	Inicializace proměnných	199
9.6.3	Programování periferních operací na vyšší úrovni	200
9.6.4	Generické jednotky	200
9.7	Volba jazykových prostředků	201
9.8	Obecné principy návrhu jazyka	204

ČÁST II

10	Vývoj RT-jazyků	207
11	RTL/2	210
11.1	Úvod	210
11.2	Jazyk RTL/2	211
11.2.1	Formální popis jazyka	211
11.2.2	Standardní vstup/výstup a ošetření chyb	222
11.3	Příklad užití jazyka RTL/2	224
11.4	Rozbor a hodnocení	233
12	Modula	236
12.1	Úvod	236
12.2	Popis jazyka	236
12.2.1	Sekvenční prostředky	237
12.2.2	Prostředky paralelního zpracování	242
12.3	Programování v jazyce Modula	245
12.3.1	Plánovač RT-procesů	245
12.3.2	Plánování uživatelských procesů	247
12.3.3	Sledování času	250
12.3.4	Vstup a výstup z terminálu	254
12.3.5	Prostředky dialogu operátora se systémem	258
12.3.6	Plánovač procesů	260
12.4	Implementace jazyka Modula	262
12.5	Zhodnocení	267
13	Ada	270
13.1	Vznik jazyka Ada	270
13.2	Jazyk Ada	272

13.2.1	Základní informace	272
13.2.2	Jednoduché datové typy	274
13.2.3	Strukturované datové typy	279
13.2.4	Klasické řídicí struktury	284
13.2.5	Mechanismy datové abstrakce	288
13.2.6	Paralelismus	292
13.2.7	Přímé programování periferních operací	296
13.2.8	Obsluha výjimek	298
13.2.9	Separátní kompilace	300
13.2.10	Vstup a výstup	302
13.3	Užití jazyka Ada	303
13.3.1	Systém dálkového sběru dat	303
13.3.2	Specifikace	305
13.3.3	Implementace	312
13.4	Celkové zhodnoení	328
	Literatura	332
	Rejstřík	335