

# Obsah

<b>Předmluva k aktuálnímu vydání</b> .....	16
<b>Předmluva k prvnímu vydání prvního dílu</b> .....	18
<b>1 Úvod</b> .....	20
1.1 C – vznik, vývoj, charakteristika .....	20
1.2 Typografické a syntaktické konvence .....	21
1.3 Styl psaní programů .....	22
<b>2 Základní pojmy</b> .....	23
2.1 Způsob zpracování programu .....	23
2.2 Základní pojmy v jazyce C .....	24
2.2.1 Zdrojové a hlavičkové soubory .....	25
2.2.2 Bílé znaky .....	26
2.2.3 ASCII tabulka .....	26
2.2.4 Identifikátory .....	27
2.2.5 Komentáře .....	28
<b>3 První začátky s C</b> .....	29
3.1 Jednoduché datové typy a přiřazení .....	29
3.1.1 Definice proměnných .....	30
3.1.2 Přiřazení .....	31
3.2 Hlavní program .....	31
3.3 Konstanty .....	33
3.3.1 Celočíselné konstanty .....	33
3.3.2 Reálné konstanty .....	34
3.3.3 Znakové konstanty .....	34
3.3.4 Řetězcové konstanty (literály) .....	35
3.4 Aritmetické výrazy .....	35
3.4.1 Unární operátory .....	35
3.4.2 Binární operátory .....	36
3.4.3 Speciální unární operátory .....	36
3.4.4 Přiřazovací operátory .....	37

<b>4 Terminálový vstup a výstup</b>	39
4.1 Hlavičkový soubor <code>stdio.h</code>	39
4.2 Vstup a výstup znaku	39
4.3 Formátovaný vstup a výstup	40
4.3.1 Řídící řetězec formátu	41
<b>5 Řídící struktury</b>	46
5.1 Booleovské výrazy	48
5.1.1 Zkrácené vyhodnocování logických výrazů	48
5.1.2 Priority vyhodnocování výrazů	49
5.2 Podmíněný výraz – ternární operátor	50
5.3 Operátor čárky	51
5.4 Příkaz <code>if</code> a příkaz <code>if--else</code>	52
5.5 Iterační příkazy – cykly	55
5.5.1 Příkazy <code>break</code> a <code>continue</code>	55
5.5.2 Příkaz <code>while</code>	56
5.5.3 Příkaz <code>do--while</code>	57
5.5.4 Příkaz <code>for</code>	58
5.6 Příkaz <code>switch</code>	60
5.7 Příkaz <code>goto</code>	65
5.8 Příkaz <code>return</code>	65
<b>6 Vstup ze souboru a výstup do souboru</b>	69
6.1 Začátek práce se souborem	71
6.1.1 Otevření souboru pro čtení	72
6.1.2 Otevření souboru pro zápis	72
6.2 Základní operace s otevřeným souborem	72
6.3 Ukončení práce se souborem	73
6.4 Příklady základní práce se soubory	73
6.5 Testování konce řádky	75
6.6 Testování konce souboru	77
6.6.1 Pomocí symbolické konstanty <code>E0F</code>	77
6.6.2 Pomocí standardního makra <code>feof()</code>	78
6.7 Testování správnosti otevření a uzavření souboru	78
6.8 Standardní vstup a výstup	82
6.9 Vracení přečteného znaku zpět do vstupního bufferu	84

<b>7</b>	<b>Typová konverze</b>	87
7.1	Implicitní typová konverze	87
7.2	Explicitní typová konverze	88
<b>8</b>	<b>Preprocesor jazyka C</b>	90
8.1	Makra bez parametrů – příkaz <code>define</code>	91
8.2	Makra s parametry	94
8.2.1	Předdefinovaná makra	95
8.3	Vkládání souborů – příkaz <code>include</code>	97
8.3.1	Vkládané soubory	97
8.3.2	Standardní hlavičkové soubory	98
8.4	Oddělený překlad souborů – I.	99
8.5	Podmíněný překlad	100
8.5.1	Řízení překladu hodnotou konstantního výrazu	101
8.5.2	Řízení překladu definicí makra	103
8.5.3	Operátor <code>defined</code>	104
8.5.4	Direktivy <code>#elif</code> a <code>#error</code>	104
<b>9</b>	<b>Funkce a práce s pamětí</b>	107
9.1	Alokace paměti	108
9.1.1	Statická alokace	108
9.1.2	Dynamická alokace na hromadě	109
9.1.3	Dynamická alokace v zásobníku	109
9.2	Funkce	110
9.2.1	Definice funkce	110
9.2.2	Procedury a datový typ <code>void</code>	112
9.2.3	Rekurzivní funkce	113
9.2.4	Funkce nevracející <code>int</code>	113
9.2.5	Problémy s umístěním definice funkcí	113
9.2.6	Konverze návratové hodnoty funkce	116
9.2.7	Parametry funkcí	116
	Konverze skutečných parametrů	117
9.3	Oblast platnosti identifikátorů	117
9.3.1	Globální a lokální proměnné	117
9.3.2	Paměťové třídy	121
	Třída <code>auto</code>	121
	Třída <code>extern</code>	122
	Třída <code>static</code>	122
	Třída <code>register</code>	123

9.3.3	Typové modifikátory .....	124
Modifikátor <code>const</code> .....	124	
Modifikátor <code>volatile</code> .....	125	
9.3.4	Bloky .....	126
9.4	Oddělený překlad souborů – II. ....	127
9.4.1	Rozšíření platnosti globální proměnné .....	127
9.4.2	Statické globální proměnné a funkce .....	128
9.4.3	Jak udržet pořádek ve velkém programu .....	130
Doporučený obsah <code>.c</code> souboru .....	131	
Doporučený obsah <code>.h</code> souboru .....	132	
9.5	Inicializace jednoduchých proměnných .....	140
<b>10</b>	<b>Pointery</b> .....	<b>144</b>
10.1	Základy práce s pointery .....	145
10.1.1	Definice dat typu <code>pointer</code> na typ .....	145
10.1.2	Práce s adresovými operátory .....	146
10.1.3	Přiřazení hodnoty pointerům a pomocí pointerů .....	146
10.1.4	Použití pointerů v přiřazovacích příkazech .....	147
10.1.5	Nulový <code>pointer</code> <code>NULL</code> .....	149
10.1.6	Konverze pointerů .....	150
10.1.7	Zarovnávaní v paměti .....	150
10.2	Pointery a funkce .....	150
10.2.1	Volání odkazem .....	151
10.2.2	<code>Pointer</code> na typ <code>void</code> .....	154
<code>Pointer</code> na typ <code>void</code> jako <code>pointer</code> na několik různých typů .....	154	
<code>Pointer</code> na typ <code>void</code> jako formální parametr funkce .....	155	
10.2.3	Pointery na funkce a funkce jako parametry funkcí ...	155
10.3	Jak číst komplikované definice – I. ....	158
10.4	Definice s využitím operátoru <code>typedef</code> .....	159
10.5	Pointerová aritmetika .....	160
10.5.1	Operátor <code>sizeof</code> .....	161
10.5.2	Součet pointeru a celého čísla .....	161
10.5.3	Odečítání celého čísla od pointeru .....	163
10.5.4	Porovnávání pointerů .....	163
10.5.5	Odečítání pointerů .....	164
10.6	Dynamické přidělování a navracení paměti .....	164
10.6.1	Přidělení paměti .....	165
10.6.2	Uvolňování paměti .....	167
10.6.3	Příklady přidělování paměti .....	167

10.6.4	Funkce <code>calloc()</code> .....	168
10.7	Pointer jako skutečný parametr funkce .....	169
<b>11</b>	<b>Jednorozměrná pole</b> .....	<b>173</b>
11.1	Základní dovednosti .....	173
11.2	Pole a pointery .....	176
11.2.1	Dynamická pole .....	177
11.2.2	Podobnost statických a dynamických polí .....	178
11.2.3	Další zvláštnosti a dovednosti při práci s poli .....	179
	Práce s celým polem najednou .....	179
	Přístup do pole pomocí pointerů .....	179
	Jak zjistit velikost pole .....	181
11.3	Pole měnící svoji velikost .....	181
11.4	Pole jako parametry funkcí .....	183
11.5	Pole pointerů na funkce .....	187
11.6	Jak číst komplikované definice – II. ....	188
<b>12</b>	<b>Řetězce</b> .....	<b>192</b>
12.1	Základní informace a definování řetězců .....	192
12.2	Práce s řetězcem .....	196
12.2.1	Čtení řetězce z klávesnice .....	196
	Čtení řetězce v daném formátu .....	196
12.2.2	Tisk řetězce na obrazovku .....	198
12.2.3	Přístup k jednotlivým znakům řetězce .....	199
12.2.4	Standardní funkce pro práci s řetězci .....	200
	Délka řetězce .....	200
	Kopírování řetězce .....	200
	Spojení řetězců .....	200
	Nalezení znaku v řetězci .....	200
	Porovnání dvou řetězců .....	200
	Nalezení podřetězce v řetězci .....	201
	Práce s omezenou částí řetězce .....	201
	Práce s řetězcem pozpátku .....	201
	Převody řetězců na čísla .....	201
12.3	Formátované čtení a zápis z a do řetězce .....	202
12.4	Řádkově orientovaný vstup a výstup z terminálu .....	204
12.4.1	Čtení řádky z klávesnice .....	204
12.4.2	Výpis řádky na obrazovku .....	205
12.5	Řádkově orientovaný vstup a výstup ze souboru .....	205

12.5.1	Čtení řádky ze souboru .....	206
12.5.2	Zápis řádky do souboru .....	208
12.6	Řídící řetězec formátu pro tisk .....	209
12.6.1	<i>konverze</i> .....	209
12.6.2	<i>modifikátor</i> .....	210
12.6.3	<i>šířka</i> .....	210
12.6.4	<i>přesnost</i> .....	211
12.6.5	<i>příznak</i> .....	211
12.6.6	Příklady různých formátů tisku .....	212
<b>13</b>	<b>Vícerozměrná pole</b> .....	215
13.1	Základní definice a přístup k prvkům .....	215
13.2	Uložení vícerozměrných polí v paměti .....	216
13.3	Různé způsoby definice dvourozměrných polí .....	218
13.3.1	Statické dvourozměrné pole .....	218
13.3.2	Pole pointerů .....	218
13.3.3	Pointer na pole .....	219
13.3.4	Pointer na pointer .....	220
13.3.5	Výhody a nevýhody předchozích čtyř způsobů .....	220
13.3.6	Dvourozměrné pole jako parametr funkce .....	222
13.4	Inicializace polí všech rozměrů .....	224
13.5	Pole řetězců .....	225
13.6	Parametry funkce <code>main()</code> .....	227
13.7	Externí pole všech rozměrů .....	229
<b>14</b>	<b>Struktury, uniony a výčtové typy</b> .....	232
14.1	Struktury .....	232
14.1.1	Definice a základní dovednosti .....	232
14.1.2	Struktury a pointery .....	235
14.1.3	Struktury odkazující samy na sebe .....	236
14.1.4	Struktura v jiné struktuře .....	238
14.1.5	Položky struktury jsou pointery .....	240
14.1.6	Alokace paměti pro jednotlivé položky struktury .....	244
14.1.7	Struktury a funkce .....	245
14.1.8	Shrnutí poznatků o práci s jednotlivými strukturami .....	248
14.1.9	Pole struktur .....	249
14.1.10	Inicializace struktur .....	252
14.2	Výčtový typ .....	253
14.3	Uniony .....	255

<b>15 Bitové operace a bitové pole</b>	262
15.1 Operace s jednotlivými bity	262
15.1.1 Bitový součin	263
15.1.2 Bitový součet	263
15.1.3 Bitový exkluzivní součet	263
15.1.4 Operace bitového posunu doleva	264
15.1.5 Operace bitového posunu doprava	264
15.1.6 Negace bit po bitu	265
15.1.7 Způsoby práce se skupinou bitů	266
15.2 Bitové pole	266
<b>16 Tabulka preferencí</b>	269

## Obsah druhého dílu

(je uveden pro úplnost – podrobnosti viz v předmluvě)

<b>17 Komplexní pohled na souborový vstup a výstup</b>	272
17.1 Rozdíly mezi binárním a textovým souborem obecně	273
17.1.1 Binární soubory	273
17.1.2 Textové soubory	274
17.2 Rozdíly mezi binárním a textovým režimem otevírání souboru	274
17.3 Otevření souboru	277
17.3.1 Různé způsoby otevírání souboru	278
17.4 Uzavření souboru	281
17.5 Ošetření chyb	283
17.6 Čtení a zápis dat	285
17.6.1 Formátované vstupy a výstupy	286
17.6.2 Neformátované vstupy a výstupy	286
Zpracování jednoho znaku v jeden okamžik	287
Zpracování celé řádky najednou	290
Zpracování celého bloku dat najednou	292
17.7 Bufferování	293
17.7.1 Základní informace	294
17.7.2 Možnosti bufferování poskytnuté ANSI C	295
17.7.3 Funkce pro práci s bufferem	295
Funkce <code>setbuf()</code>	295
Funkce <code>setvbuf()</code>	296
17.7.4 Řešení problému nebufferovaného <code>stdin</code>	297

17.8	Funkce pro zápis bufferu na disk .....	299
17.8.1	Funkce <code>fflush()</code> .....	299
17.9	Přímý přístup do souboru .....	300
17.9.1	Funkce <code>fseek()</code> – posun v souboru .....	300
17.9.2	Funkce <code>ftell()</code> – zjištění pozice v souboru .....	300
17.9.3	Typické použití přímého přístupu .....	302
17.10	Zbývající užitečné funkce ze <code>&lt;stdio.h&gt;</code> .....	306
17.10.1	Funkce <code>freopen()</code> – přesměrování proudu .....	306
17.10.2	Funkce <code>rename()</code> – přejmenování souboru .....	308
17.10.3	Funkce <code>remove()</code> – zrušení souboru .....	308
17.10.4	Funkce <code>tmpfile()</code> – pomocný soubor .....	308
17.10.5	Funkce <code>tmpnam()</code> – jméno pomocného souboru .....	308
17.11	Ukázka použití funkcí z UNIXové knihovny .....	309
17.12	Zjištění informací o položkách v adresáři .....	311
17.12.1	Prostředí UNIXu .....	311
	Zjištění pouze jmen položek .....	312
	Zjištění typu a stavu položek .....	313
17.12.2	Prostředí Windows .....	316
<b>18</b>	<b>Návaznost překladače ANSI C na okolní prostředí .....</b>	<b>319</b>
18.1	Standardní hlavičkové soubory .....	319
18.2	Rezervovaná jména .....	321
18.3	Standardní hlavičkové soubory bez funkčních prototypů .....	322
18.3.1	Soubor <code>&lt;errno.h&gt;</code> .....	322
18.3.2	Soubor <code>&lt;limits.h&gt;</code> .....	323
18.3.3	Soubor <code>&lt;float.h&gt;</code> .....	323
18.3.4	Soubor <code>&lt;stddef.h&gt;</code> .....	324
18.4	Předdefinovaná makra preprocesoru .....	325
18.4.1	Makro <code>__STDC__</code> .....	325
18.4.2	Makro <code>__FILE__</code> .....	325
18.4.3	Makro <code>__LINE__</code> .....	325
18.4.4	Makro <code>__DATE__</code> .....	325
18.4.5	Makro <code>__TIME__</code> .....	326
18.5	Externí proměnné .....	326
18.5.1	Proměnná <code>errno</code> z <code>&lt;errno.h&gt;</code> .....	326
18.5.2	Proměnná <code>sys_errlist</code> ze <code>&lt;stdlib.h&gt;</code> .....	327
18.5.3	Proměnná <code>sys_nerr</code> ze <code>&lt;stdlib.h&gt;</code> .....	327
18.5.4	Proměnná <code>environ</code> ze <code>&lt;stdlib.h&gt;</code> .....	328
18.5.5	Proměnná <code>_iob</code> ze <code>&lt;stdio.h&gt;</code> .....	328



<b>19 Popis funkcí a maker ze standardních knihoven</b>	<b>329</b>
19.1 <assert.h> – makro používané při ladění	329
19.2 <ctype.h> – práce se znaky	329
19.3 <math.h> – matematické funkce v přesnosti double	330
19.4 <locale.h> – přizpůsobení C národnímu prostředí	331
19.5 <setjmp.h> – umožnění běžně nedovolených skoků	332
19.6 <signal.h> – zpracování signálů	334
19.7 <stdarg.h> – práce s proměnným počtem parametrů	337
19.8 <stdio.h> – funkce pro vstup a výstup	337
19.9 <stdlib.h> – obecně užitečné funkce	339
19.9.1 Konverze řetězců na čísla	339
Funkce pro jednoduchý převod desítkových čísel	339
Funkce pro náročný převod čísel	339
19.9.2 Generátor pseudonáhodných čísel	340
19.9.3 Funkce pracující s dynamickou pamětí	341
19.9.4 Funkce pro spolupráci s operačním systémem	341
Funkce <code>abort()</code>	341
Funkce <code>atexit()</code>	341
Funkce <code>exit()</code>	342
Funkce <code>system()</code>	343
19.9.5 Funkce pro hledání a řazení	343
Řazení algoritmem <i>quick-sort</i>	343
Binární vyhledávání v seřazeném poli	344
19.10 <string.h> – zpracování řetězců	348
19.10.1 Funkce <code>strspn()</code>	348
19.10.2 Funkce <code>strcspn()</code>	349
19.10.3 Funkce <code>strpbrk()</code>	349
19.10.4 Funkce <code>strtok()</code>	349
19.10.5 Funkce <code>memchr()</code>	350
19.10.6 Funkce <code>memcmp()</code>	350
19.10.7 Funkce <code>memcpy()</code>	351
19.10.8 Funkce <code>memmove()</code>	351
19.10.9 Funkce <code>memset()</code>	351
19.11 <time.h> – práce s datem a časem	353
19.11.1 Funkce <code>clock()</code>	353
19.11.2 Měření doby běhu programu	354
19.11.3 Funkce <code>time()</code>	356
19.11.4 Funkce <code>difftime()</code>	356

19.11.5	Funkce <code>localtime()</code> .....	356
19.11.6	Funkce <code>asctime()</code> .....	357
19.11.7	Funkce <code>ctime()</code> .....	357
19.11.8	Funkce <code>strftime()</code> .....	358
19.11.9	Funkce <code>mktime()</code> .....	359
<b>20</b>	<b>Funkce s proměnným počtem parametrů .....</b>	<b>362</b>
20.1	Úvodní informace .....	362
20.2	Základní princip .....	363
20.3	Standardní hlavičkový soubor <code>&lt;stdarg.h&gt;</code> .....	365
20.4	Funkce <code>vprintf()</code> , <code>vfprintf()</code> , <code>vsprintf()</code> .....	366
20.5	Praktické příklady použití FPPP .....	367
20.5.1	Typické použití <code>vprintf()</code> , <code>vfprintf()</code> , <code>vsprintf()</code> .....	367
20.5.2	Přesměrování výstupu za běhu programu .....	368
20.5.3	Seznam parametrů je ukončený domluvenou hodnotou .....	370
20.5.4	Počet parametrů je předán v pevném parametru .....	372
20.5.5	Odstrašující případ zcela netypického použití .....	373
<b>21</b>	<b>Ladění v jazyce C .....</b>	<b>376</b>
21.1	Ladicí výpisy na <code>stderr</code> .....	378
21.2	Využití podmíněného překladu pomocí příkazů C-preprocesoru .....	378
21.2.1	Využití existence symbolické konstanty .....	379
21.2.2	Využití hodnoty symbolické konstanty .....	380
21.3	Self-testy modulů nebo funkcí .....	383
21.4	Využití makra ze souboru <code>&lt;assert.h&gt;</code> .....	385
21.4.1	Praktické použití .....	386
21.4.2	Zrušení aserce .....	389
21.4.3	Princip aserce .....	389
21.4.4	Chybné použití aserce .....	390
21.5	Prostředky pro ladění dynamických procesů .....	392
<b>22</b>	<b>Minimální znalosti při programování pod UNIXem .....</b>	<b>396</b>
22.1	Editor .....	396
22.2	Překladač a sestavovací program .....	397
22.3	Kontrola zdrojových textů v C .....	399
22.4	Odděleně překládané moduly .....	401
22.5	Ladicí prostředky .....	404

<b>23</b>	<b>Změny v C podle nového standardu ISO</b>	406
23.1	Stručný přehled norem a nejdůležitějších změn	406
23.2	Milé drobnosti	407
23.2.1	Komentáře do konce řádky	407
23.2.2	Definice kdekoliv v kódu	408
23.2.3	Co bylo zrušeno	408
23.3	Celočíselné datové typy	409
23.3.1	Logické hodnoty	409
23.3.2	Typ <code>long long int</code>	410
23.3.3	Celočíselné typy s přesně danou šířkou	410
23.3.4	Nové formátové specifikace	412
23.4	Okrajové hodnoty reálných čísel	413
23.4.1	Nekonečno	413
23.4.2	NaN	413
23.4.3	Denormální čísla	413
23.4.4	Podpora okrajových hodnot reálných čísel	413
23.5	Pole	414
23.5.1	Inicializace polí	414
23.5.2	Proměnná délka pole	415
23.5.3	Minimální velikost pole jako parametr funkce	416
23.5.4	Složené literály	416
23.6	Funkce	416
23.6.1	Identifikátor funkce	417
23.6.2	Funkce mohou být vnořené	418
23.6.3	Modifikátor <code>inline</code>	419
23.7	Vícebajtové znaky	420
23.7.1	Kódování a definice proměnných	421
23.7.2	Práce s jednotlivými znaky	421
23.7.3	Práce s řetězci	422
23.7.4	Vstupy a výstupy	423
<b>24</b>	<b>Desatero příkázání pro programátory v C</b>	425
	<b>Literatura</b>	430
	<b>Rejstřík</b>	431