

BRIAN W. KERNIGHAN
DENNIS M. RITCHIE

PROGRAMOVACÍ JAZYK

ANSI C99



OD AUTORŮ JAZYKA C

ZÁKLADNÍ GRAMATIKA JAZYKA
STAVBA A ŘÍZENÍ BĚHU PROGRAMU
POLE, UKAZATELE, STRUKTURY
VSTUP/VÝSTUP, OŠETŘENÍ CHYB
STANDARDNÍ KNIHOVNA
NOVINKY A ZMĚNY DLE STANDARDU C99



knihy.cpress.cz

Brian W. Kernighan, Dennis M. Ritchie

Programovací jazyk C

**Computer Press
Brno
2013**

Programovací jazyk C

Brian W. Kernighan, Dennis M. Ritchie

Překlad: Zbyněk Šáva

Odborná korektura: Miroslav Virius

Obálka: Martin Sodomka

Odpovědný redaktor: Martin Domes

Technický redaktor: Jiří Matoušek

Authorized translation from the English language edition, entitled „C PROGRAMMING LANGUAGE, 2 nd Edition“, ISBN 0131103628, by KERNINGHAN, BRIAN W.; RITCHIE, DENNIS, published by Pearson Education, Inc, publishing as Prentice Hall PTR, Copyright © 1988 All rights reserved.

No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from the Publisher. CZECH language edition published by Computer Press, a.s., Copyright © 2006.

Objednávky knih:

<http://knihy.cpress.cz>

www.albatrosmedia.cz

eshop@albatrosmedia.cz

bezplatná linka 800 555 513

ISBN 978-80-251-0897-0

Vydalo nakladatelství Computer Press v Brně roku 2013 ve společnosti Albatros Media a.s. se sídlem Na Pankráci 30, Praha 4. Číslo publikace 16695.

© Albatros Media a.s. Všechna práva vyhrazena. Žádná část této publikace nesmí být kopírována a rozmnožována za účelem rozšiřování v jakékoli formě či jakýmkoli způsobem bez písemného souhlasu vydavatele.

Dotisk 1. vydání

 **ALBATROS** MEDIA a.s.

Obsah

Předmluva k českému vydání	11
Předmluva	13
Předmluva k prvnímu vydání	15
Úvod	17

Kapitola 1

Úvodní kurz **21**

1.1	Začínáme	21
1.2	Proměnné a aritmetické výrazy	24
1.3	Příkaz for	28
1.4	Symbolické konstanty	29
1.5	Znakový vstup a výstup	30
1.5.1	Kopírování souboru	30
1.5.2	Počítání znaků	32
1.5.3	Počítání řádků	33
1.5.4	Počítání slov	34
1.6	Pole	35
1.7	Funkce	37
1.8	Argumenty – předávání hodnotou	40
1.9	Znaková pole	41
1.10	Externí proměnné a oblast platnosti	43
1.11	Standard C99	46

Kapitola 2

Typy, operátory a výrazy **47**

2.1	Jména proměnných	47
2.2	Datové typy a velikosti	47
2.3	Konstanty	48
2.4	Deklarace	51
2.5	Aritmetické operátory	52
2.6	Relační a logické operátory	52
2.7	Konverze typů	53
2.8	Operátory inkrementace a dekrementace	57
2.9	Bitové operátory	58
2.10	Přiřazovací operátory a výrazy	60

2.11	Podmíněné výrazy	61
2.12	Priorita a pořadí výpočtu	62
2.12	Standard C99	63
2.12.1	Celočíselné typy	64
2.12.2	Čísla s pohyblivou řádovou čárkou	67
2.12.3	Komplexní čísla	69
2.12.4	Konverze	70

Kapitola 3

Řízení běhu programu 73

3.1	Příkazy a bloky	73
3.2	If-else	73
3.3	Else-if	74
3.4	Switch	76
3.5	Cykly – while a for	77
3.6	Cykly – do-while	80
3.7	Break a continue	81
3.8	goto a návěští	82
3.9	Standard C99	83
3.9.1	Bloky a deklarace	83
3.9.2	Cykly	83
3.9.3	Výběrové (podmíněné) příkazy	84
3.9.4	Skok	84

Kapitola 4

Funkce a struktura programu 85

4.1	Začínáme s funkcemi	85
4.2	Funkce, které nevrací celá čísla	88
4.3	Externí proměnné	90
4.4	Pravidla rozsahu platnosti	96
4.5	Hlavičkové soubory	97
4.6	Statické proměnné	98
4.7	Registrové proměnné	99
4.8	Bloková struktura	100
4.9	Inicializace	100
4.10	Rekurze	101
4.11	Preprocesor jazyka C	103
4.11.1	Vkládání souborů	103
4.11.2	Substituce maker	104
4.11.3	Podmíněný překlad	105
4.12	Standard C99	106

4.12.1	Funkce	106
4.12.2	Makra	108
4.12.3	#pragma	108

Kapitola 5

Ukazatele a pole **111**

5.1	Ukazatele a adresy	111
5.2	Ukazatele a argumenty funkcí	113
5.3	Ukazatele a pole	115
5.4	Adresová aritmetika	117
5.5	Funkce a ukazatele na znaky	120
5.6	Ukazatele na pole; ukazatele na ukazatele	123
5.7	Vícerozměrná pole	126
5.8	Inicializace polí ukazatelů	128
5.9	Ukazatele versus vícerozměrná pole	128
5.10	Argumenty příkazové řádky	129
5.11	Ukazatele na funkce	133
5.12	Komplikované deklarace	136
5.13	Standard v C99	140
5.13.1	Restrigované (omezené) ukazatele	140
5.13.2	Pole	142
5.13.3	Pole jako parametr funkce	143

Kapitola 6

Struktury **145**

6.1	Základní informace o strukturách	145
6.2	Struktury a funkce	147
6.3	Pole struktur	149
6.4	Ukazatele na struktury	153
6.5	Struktury odkazující na sebe	155
6.6	Vyhledávání v tabulkách	159
6.7	Typedef	161
6.9	Unie	162
6.9	Bitová pole	163
6.10	Standard C99	165
6.10.1	Bitová pole	165
6.10.2	Inicializace struktur a unií	165
6.10.3	Literály typu struktura a unie	165

Kapitola 7

Vstup a výstup	167
7.1	Standardní vstup a výstup 167
7.2	Formátovaný výstup – funkce printf 169
7.3	Seznamy argumentů proměnné délky 171
7.4	Formátovaný vstup – funkce scanf 172
7.5	Přístup k souborům 175
7.6	Ošetření chyb – funkce stderr a exit 177
7.7	Vstup a výstup po řádcích 179
7.8	Různé funkce 180
7.8.1	Operace s řetězci 180
7.8.2	Testování tříd znaků a konverze 180
7.8.3	Funkce ungetc 181
7.8.4	Vykonání příkazu 181
7.8.5	Správa paměti 181
7.8.6	Matematické funkce 182
7.8.7	Generování náhodných čísel 182
7.9	Standard C99 182
7.9.1	Datové proudy 183

Kapitola 8

Rozhraní systému UNIX	185
8.1	Deskriptory souborů 185
8.2	Nízkoúrovňový vstup a výstup 186
8.3	Open, creat, close, unlink 187
8.4	Náhodný přístup – lseek 190
8.5	Příklad – implementace funkcí fopen a getc 190
8.6	Příklad – výpis adresářů 194
8.7	Příklad – alokátor paměti 199

Příloha A

Referenční příručka	203
A1.	Úvod 203
A2.	Lexikální konvence 203
A2.1	Symboly 203
A2.2	Komentáře 203
A2.3	Identifikátory 204
A2.4	Klíčová slova 204
A2.5	Konstanty 204
A2.6	Řetězcové literály 206

A3	Zápis syntaxe	206
A4	Význam identifikátorů	207
A4.1	Paměťová třída	207
A4.2	Základní typy	207
A4.3	Odvozené typy	208
A4.4	Kvalifikátory typů	208
A5	Objekty a l-hodnoty	209
A6	Konverze	209
A6.1	Celočíselná rozšíření	209
A6.2	Celočíselné konverze	209
A6.3	Celá čísla a čísla s pohyblivou řádovou čárkou	209
A6.4	Typy s pohyblivou řádovou čárkou	210
A6.5	Aritmetické konverze	210
A6.6	Ukazatele a celá čísla	210
A6.7	Void	211
A6.8	Ukazatele na void	211
A7	Výrazy	212
A7.1	Vytváření ukazatelů	212
A7.2	Primární výrazy	212
A7.3	Postfixové výrazy	213
A7.4	Unární operátory	215
A7.5	Přetypování	217
A7.6	Multiplikativní operátory	217
A7.7	Aditivní operátory	217
A7.8	Operátory posunu	218
A7.9	Relační operátory	218
A7.10	Operátory rovnosti	219
A7.11	Operátor bitové konjunkce	219
A7.12	Operátor bitové nonekvivalence	219
A7.13	Operátor bitové disjunkce	219
A7.14	Operátor logické konjunkce	219
A7.15	Operátor logické disjunkce	220
A7.16	Podmíněný operátor	220
A7.17	Výrazy přiřazení	220
A7.18	Operátor čárka	221
A7.19	Konstantní výrazy	221
A8	Deklarace	222
A8.1	Specifikátory paměťové třídy	222
A8.2	Specifikátory typů	223
A8.3	Deklarace struktur a unií	224
A8.4	Výčty	227
A8.5	Deklarace	228

	A8.6 Význam deklarátorů	228
	A8.7 Inicializace	231
	A8.8 Jména typů	233
	A8.9 Typedef	234
	A8.10 Ekvivalence typů	234
A9	Příkazy	235
	A9.1 Příkazy s návěštím	235
	A9.2 Výrazový příkaz	235
	A9.3 Složený příkaz	235
	A9.4 Výběrové příkazy	236
	A9.5 Iterační příkazy	237
	A9.6 Příkazy skoku	237
A10	Externí deklarace	238
	A10.1 Definice funkcí	238
	A10.2 Externí deklarace	239
A11	Rozsah platnosti a vazba	240
	A11.1 Lexikální rozsah platnosti	240
	A11.2 Vazba	241
A12	Preprocesor	241
	A12.1 Trigrafy	242
	A12.2 Spojování řádků	242
	A12.3 Definice a rozvoj maker	242
	A12.4 Vkládání souborů	244
	A12.5 Podmíněná kompilace	245
	A12.6 Řízení řádků	246
	A12.7 Generování chyb	246
	A12.8 Pragma	246
	A12.9 Prázdná direktiva	247
	A12.10 Předdefinovaná jména	247
A13	Gramatika	247

Příloha B

Standardní knihovna

255

B1.	Vstup a výstup: <stdio.h>	255
	B1.1 Operace se soubory	256
	B1.2 Formátovaný výstup	257
	B1.3 Formátovaný vstup	259
	B1.4 Funkce pro vstup a výstup po jednotlivých znacích	261
	B1.5 Funkce pro přímý vstup a výstup	262
	B1.6 Funkce pracující s pozicí v souboru	262
	B1.7 Chybové funkce	262

B2.	Testy tříd znaků: <ctype.h>	263
B3.	Funkce pracující s řetězci <string.h>	264
B4.	Matematické funkce: <math.h>	265
B5.	Užitečné funkce: <stdlib.h>	266
B6.	Ladění: <assert.h>	269
B7.	Seznam argumentů proměnné délky: <stdarg.h>	269
B8.	Nelokální skoky: <setjmp.h>	270
B9.	Signály: <signal.h>	270
B10.	Funkce pro práci s datem a časem: <time.h>	271
B11.	Implementací definované meze: <limits.h> a <float.h>	273
Příloha C		
Shrnutí změn		275
Příloha D		
Hlavní novinky standardu C99		279
Rejstřík		281

Předmluva k českému vydání

Držíte v rukou nejznámější knihu o jazyce C, jaká kdy byla napsána – knihu nejen stále aktuální, ale v mnoha ohledech také stále nepřekonanou. Jedním z jejích autorů je Denis Ritchie, který v roce 1972 navrhl a implementoval první verzi jazyka C; spolu s Brianem W. Kernighanem pak v roce 1978 vydali knihu *The C Programming Language*, která se na dlouhou dobu stala neoficiálním standardem tohoto jazyka.

Jazyk popsáný v prvním vydání této knihy se dodnes označuje jako „jazyk C podle Kernighana a Ritchieho“, případně „C podle K&R“, a s jeho implementacemi se lze stále ještě setkat. U nás je toto první vydání známo ze slovenského překladu vydaného nakladatelstvím Alfa (Bratislava, 1988).

V roce 1988 vyšlo druhé, aktualizované vydání, které popisuje tehdy připravovaný standard ANSI X3.159-1989. Překlad tohoto vydání se vám nyní dostává do rukou.

Americký národní standard jazyka C byl v USA v roce 1990 stažen a nahrazen mezinárodním standardem ISO/IEC 9899-1990, dnes běžně označovaným jako *C90*. To nic nemění na skutečnosti, že se američtí výrobci softwaru stále odvolávají na standard ANSI. Obrátíte-li se na Americký národní standardizační institut, ANSI, prodá vám jako standard jazyka C zmíněnou normu ISO.

Dnešní překladače jazyka C zpravidla plně vyhovují standardu *C90*.

V roce 1999 byla přijata nová verze standardu jazyka C, dnes označovaná jako *C99*. Ta přinesla řadu úprav a rozšíření, o nichž se dozvíte v dodatcích k jednotlivým kapitolám, nadepsaných *Standard C99*; jejich stručný souhrn pak najdete v dodatku D. Současné překladače přistupují ke standardu *C99* zatím opatrně: většinou implementují pouze některé z novinek. To se však v dohledné době může změnit.

Při překladu této knihy jsme zachovali původní text, nesnažili jsme se o úpravu podle standardu *C99*; pouze na místa, která by mohla při překladu v *C99* způsobit problémy, jsme vložili upozornění v podobě poznámek pod čarou, na závěr většiny kapitol jsme vložili oddíl *Standard C99*, v němž jsou shrnuty novinky a změny, a na závěr knihy jsme připojili Přílohu D shrnující nejdůležitější změny, které standard *C99* přinesl.

Spolupracovat na překladu této knihy pro mne bylo opravdu potěšením, a proto bych rád poděkoval těm, kteří mi na přelomu 80. a 90. let pomohli tento krásný programovací jazyk zvládnout. Pracoval jsem v oné době jako odborný asistent na Katedře matematiky FJFI ČVUT na svém prvním projektu v jazyce C. Osobní počítače byly tehdy k dispozici pouze ve studovně a Ivo Majetič, který právě dokončoval program ke své diplomové práci, si našel čas a pomohl mi zorientovat se nejen v novinkách jazyka, které nebyly popsány v prvním vydání Kernighana a Ritchieho, ale především v knihovnách tohoto jazyka.

Poté, co Ivo úspěšně dostudoval, mi s jazykem C pomáhal další z tehdejších studentů, Mirek Minárik, který mne naučil luštit disasemblované programy a spolu se mnou hledal chyby jednoho z tehdy populárních překladačů jazyka C a C++.

Oběma jim patří dík.

Miroslav Virius

Katedra softwarového inženýrství FJFI ČVUT

Předmluva

Od vydání *The C Programming Language* v roce 1978 prošel svět výpočetní techniky revolucí. Velké počítače ještě nabraly na velikosti a osobní počítače disponují schopnostmi, které mohou směle soupeřit se sálovými počítači uplynulé dekády. Během této doby se změnil i programovací jazyk C (i když jen mírně) a rozšířil se daleko mimo své původní působíště – operační systém UNIX.

Rostoucí popularita jazyka C, jeho změny v uplynulých letech a vytvoření kompilátorů skupinami, které se nepodílely na jeho návrhu, jsou důvodem, proč je nutná precizní a aktuálnější definice jazyka, než jakou poskytlo první vydání této knihy. V roce 1983 sestavila organizace American National Standards Institute (ANSI) komisi, jejímž úkolem bylo vytvořit „bezspornou a strojově nezávislou definici jazyka C“, která by zachovávala původní myšlenky jazyka. Výsledkem byl standard ANSI jazyka C.

Standard formalizuje konstrukce naznačené ale nepopsané v prvním vydání, zejména výčty a přiřazování struktur. Přináší nový způsob deklarace funkcí, jež umožňuje provádět křížovou kontrolu definice funkce a jejího použití. Specifikuje standardní knihovnu s rozsáhlou množinou funkcí pro práci se vstupy a výstupy, správou paměti, manipulaci s řetězci a podobné úkoly. Přesně určuje chování vlastností, jež nebylo detailně vysvětleno v původní definici, a současně explicitně jmenuje aspekty jazyka, které zůstávají strojově závislé.

Toto druhé vydání *The C Programming Language* popisuje jazyk C tak, jak je definován standardem ANSI. Programy jsme se rozhodli psát výhradně v novém tvaru zápisu, i když zmiňujeme místa, kde se jazyk změnil. Většinou nedošlo k žádným podstatným změnám; nejviditelnější změnou je nový způsob deklarace a definice funkcí. Moderní kompilátory již většinu rysů standardu podporují.

Snažili jsme se zachovat stručnost prvního vydání. C není objemným jazykem, a proto mu nesvědčí objemné knihy. Zpracovali jsme na výkladu kritických vlastností jazyka, jako jsou ukazatele, jež jsou středem programování v jazyce C. Vyladili jsme původní příklady a do několika kapitol jsme dodali příklady nové. Části s komplikovanými deklaracemi jsou například rozšířeny o programy, které převádí deklarace do slov a naopak. Stejně jako dříve i nyní jsme testovali přímo všechny příklady z textu, který je ve strojově čitelné formě.

Příloha A, referenční příručka, není standardem, ale naší snahou sdělit vám klíčové základy standardu na menším prostoru. Je určena programátorům pro snadnější pochopení jazyka, ale nemůže sloužit jako definice pro autory kompilátorů – tato role po právu náleží samotnému standardu. Příloha B je shrnutím prostředků, které poskytuje standardní knihovna. Stejně jako příloha A je zamýšlena jako referenční příručka pro programátory a ne pro implementátory. Příloha C je stručným výčtem změn oproti původní verzi.

Jak jsme řekli v předmluvě k prvnímu vydání, C „slouží tím lépe, čím více rostou vaše zkušenosti s ním.“ S deseti roky nových zkušeností to cítíme stále stejně. Doufáme, že vám tato kniha pomůže naučit se jazyk C a správně ho používat v každodenní praxi.

Jsme hluboce zavázáni přátelům, kteří nám pomohli s tímto druhým vydáním. Jon Bentley, Doug Gwyn, Doug McIlroy, Peter Nelson a Rob Pike nám poskytli komentáře k téměř každé stránce původních návrhů. Za pečlivé čtení děkujeme Alovi Ahovi, Dennisi Allisonovi, Joeovi Campbellovi, G. R. Emlinovi, Karen Fortgangové, Allenovi Holubovi, Andrewovi Humemu, Davu Kristolovi, Johnu Lindermanovi, Daveovi Prosserovi, Geneovi Spaffordovi a Chrise Van Wykovi. Užitečné rady jsme dostali také od Billa Cheswicka, Marka Kernighana, Andyho Koeniga, Robin Lakeové, Toma Londona, Jima Reedse, Clovise Tonda a Petera Weinbergera. Dave Prosser nám zodpověděl mnoho otázek ohledně standardu ANSI. Pro lokální testování našich programů jsme často využívali překladač C++ Bjarne Stroustrupa a Dave Kristol nám poskytl kompilátor ANSI C pro finální testování. Se sazbou nám velice pomohl Rich Drechsler.

Upřímně děkujeme všem.

Brian W. Kernighan

Dennis M. Ritchie

Předmluva k prvnímu vydání

C je univerzální programovací jazyk, vyznačující se úspornými výrazy, moderním řízením běhu, moderními datovými strukturami a bohatou množinou operátorů. C není „jazykem vysoké úrovně“, ani „velkým“ jazykem a není specializován pro žádnou konkrétní oblast nasazení. Ale nepřítomnost omezení a jeho obecnost ho dělají vhodnějším a efektivnějším pro většinu úloh, než jiné „mocnější“ jazyky.

Jazyk C byl původně navržen a také implementován Dennisem Ritchiem na operačním systému UNIX na počítači DEC PDP-11. Operační systém, kompilátor jazyka C a prakticky všechny aplikace pro UNIX (včetně softwaru, jenž byl použit při přípravě této knihy) byly napsány v C. Produkční kompilátory existují také pro několik dalších počítačů včetně IBM System/370, Honeywell 6000 a Interdata 8/32. Avšak jazyk C není svázán s konkrétním hardwarem nebo systémem a je snadné psát programy, které budou fungovat beze změn na kterémkoli počítači podporujícím C.

Tato kniha si klade za cíl pomoci čtenáři naučit se programovat v jazyce C. Obsahuje úvodní kurz jazyka, který umožňuje novým uživatelům začít tak rychle, jak to jen jde, a dále samostatné kapitoly pro každý z důležitých rysů jazyka a referenční příručku. Většina výkladu je založena na čtení, psaní a revizi příkladů spíše než na výčtu pravidel. Ve většině případů jsou jako příklady uvedeny kompletní skutečné programy, nikoli izolované fragmenty kódu. Všechny příklady byly testovány přímo z textu, který je ve strojově čitelné formě. Kromě ukázek efektivního používání jazyka jsme se také snažili, kde to bylo možné, ilustrovat užitečné algoritmy a principy dobrého programátorského stylu a kvalitního návrhu.

Tato kniha není úvodem do programování; předpokládá jistou zkušenost se základními koncepty programování jako jsou proměnné, přiřazovací příkazy, cykly a funkce. Nicméně ani programátor začátečník by neměl mít problémy s chápáním výkladu, i když rady zkušenějšího kolegy mohou samozřejmě pomoci.

Naše zkušenosti ukázaly, že C je příjemný, expresivní a všestranný jazyk s širokým využitím. Snadno se učí a slouží tím lépe, čím více rostou vaše zkušenosti s ním. Doufáme, že tato kniha vám pomůže ho správně používat.

Této knize a naší radosti z jejího psaní velice pomohly rady a konstruktivní kritika mnoha přátel a kolegů. Zejména Mike Bianchi, Jim Blue, Stu Feldman, Doug McIlroy, Bill Rome, Bob Rosin a Larry Rosler pečlivě přečetli několik verzí této knihy. Jsme také zavázáni Alovi Ahovi, Steveovi Bournemu, Danu Dvorakovi, Chucku Haleyimu, Debbie Haleyové, Marion Harrisnové, Dicku Holtovi, Steveovi Johnsonovi, Johnu Masheyimu, Bobovi Mitzemu, Ralphovi Muhaovi, Peterovi Nelsonovi, Elliotovi Pinsonovi, Billovi Plaugerovi, Jerryemu Spivackovi, Kenovi Thompsonovi a Peterovi Weibergerovi za uži-

tečné připomínky k různým stádiím knihy a Mikeovi Leskovi a Joeovi Ossannaovi za neocenitelnou pomoc při sazbě.

Brian W. Kernighan

Dennis M. Ritchie

Úvod

C je univerzální programovací jazyk. Jeho historie je úzce spjata s operačním systémem UNIX, kde byl vyvinut, protože jak systém, tak i většina programů, které na něm běží, jsou napsány v C. Avšak jazyk sám není svázán s žádným operačním systémem nebo hardwarovou platformou; a i když byl nazýván „systémovým programovacím jazykem“, protože se hodí pro psaní kompilátorů a operačních systémů, byl stejně dobře využíván pro psaní důležitých programů v mnoha různých odvětvích.

Mnoho důležitých myšlenek jazyka C vychází z jazyka BCPL, který vyvinul Martin Richards. Vliv BCPL na C probíhal nepřímo skrze jazyk B vytvořený Kenem Thompsonem v roce 1970 pro první systém UNIX na počítači DEC PDP-7.

BCPL a B jsou „netypované“ jazyky. Naproti tomu C nabízí množství datových typů. Základní typy jsou znaky, celá čísla a čísla s pohyblivou desetinnou čárkou. Jazyk C navíc obsahuje hierarchii odvozených datových typů vytvořených pomocí ukazatelů, polí, struktur a unií. Výrazy se skládají z operátorů a operandů; jakýkoli výraz včetně přiřazení nebo volání funkce může být příkazem. Díky ukazatelům lze v jazyce C používat strojově nezávislou adresovou aritmetiku.

Jazyk C nabízí základní konstrukce pro řízení běhu, které jsou nezbytné pro správně strukturované programy: seskupování příkazů, rozhodování (`if-else`), výběr z množiny možných případů (`switch`), cykly s testem ukončení na počátku (`while`, `for`) nebo na konci (`do`) a předčasný skok z cyklu (`break`).

Funkce mohou vracet hodnoty základních typů, struktury, unie nebo ukazatele. Jakoukoli funkci lze volat rekurzivně. Lokální proměnné jsou obvykle „automatické“ a jsou znovu vytvářeny při každém zavolání funkce. Definice funkcí nesmí být vnořené, ale deklarace proměnných se řídí blokovou strukturou. Funkce programu v jazyce C mohou existovat v oddělených zdrojových souborech, které jsou kompilovány zvlášť. Proměnné mohou být viditelné jen v dané funkci, mimo funkci, ale pouze v jednom zdrojovém souboru, nebo v celém programu.

Preprocesor provádí náhradu maker v textu programu, vkládání dalších zdrojových souborů a podmíněnou kompilaci.

C je relativně „nízkourovňový“ jazyk. To není myšleno pejorativně; tím chceme říci, že C pracuje se stejnými objekty jako většina počítačů, jmenovitě se znaky, čísly a adresami. S tím vším je možno pracovat pomocí aritmetických a logických operátorů implementovaných skutečnými počítači.

C nenabízí žádné operace, které by přímo pracovaly se složenými objekty, jako jsou znakové řetězce, množiny, seznamy nebo pole. Neobsahuje žádné operace, které manipulují s celým polem nebo řetězcem, i když struktury lze kopírovat jako atomické objekty. Jazyk nedefinuje jiný nástroj pro alokaci paměti než statické definice a definice lokálních proměnných ve funkcích, které používají zásobník; není zde automatická správa paměti (`garbage collector`). Konečně, samotný jazyk C nemá žádné nástroje pro vstup a výstup; neobsahuje žádné příkazy `READ` nebo `WRITE` a žádné zabudované metody pro přístup

k souborům. Nicméně většina implementací jazyka C obsahuje pro tyto úkoly rozumně standardní sbírku funkcí.

Podobně, C nabízí pouze jednoduché, jednovláknové řízení běhu programu: testy, cykly, seskupování a podprogramy, ale ne multiprogramování, paralelní operace, synchronizaci nebo rutiny.

I když se absence některých těchto nástrojů může jevit jako zásadní nedostatek („Chcete říct, že musím zavolat funkci, abych porovnal dva znakové řetězce?“), malá velikost jazyka přináší skutečné výhody. Protože jazyk C je relativně malý, může být popsán na malém prostoru a je možné se jej rychle naučit. Programátor tak může rozumně předpokládat, že zná a chápe celý jazyk a může jej pravidelně používat.

Po mnoho let byla definicí jazyka C jeho referenční příručka – v prvním vydání *The C Programming Language*. V roce 1983 organizace American National Standards Committee (ANSI) ustanovila komisi, jejímž úkolem bylo vytvořit moderní, úplnou definici jazyka C. Výsledná definice, standard ANSI neboli „ANSI C“, byla dokončena koncem roku 1988. Moderní kompilátory už v té době podporovaly většinu rysů standardu.

Standard vychází z původní referenční příručky. Jazyk je změněn jen nepatrně; jedním z cílů standardu bylo zajistit, že většina existujících programů zůstane platná, nebo, v případě že se program stane neplatným, budou kompilátory varovat před novým chováním.

Pro většinu programátorů byla nejdůležitější změnou nová syntaxe deklarace a definice funkcí. Deklarace funkce nyní může obsahovat popis argumentů funkce; syntaxe definice se změnila stejným způsobem. Tato informace navíc velice usnadňuje kompilátorům práci při detekci chyb způsobených neodpovídajícími argumenty; podle naší zkušenosti jde o velice užitečné rozšíření jazyka.

V jazyce došlo i k jiným menším změnám. Výčty a přiřazení struktur, které patřily k běžným rozšířením, jsou nyní oficiálně součástí jazyka. Výpočty s pohyblivou desetinnou čárkou lze nyní provádět s jednoduchou přesností. Vlastnosti aritmetiky, zvláště pro typy bez znaménka, byly upřesněny. Preprocesor je propracovanější. Většina těchto změn má pouze malý vliv na většinu programátorů.

Druhým důležitým přínosem standardu je specifikace knihovny, která doprovází jazyk C. Specifikace definuje funkce pro přístup k operačnímu systému (například pro čtení ze souborů a zápis do nich), formátovaný vstup a výstup, alokaci paměti, manipulaci s řetězci a další. Sbíрка standardních hlavičkových souborů představuje jednotný přístup k deklaracím funkcí a datových typů. Programy, které používají tuto knihovnu pro komunikaci s hostitelským systémem, mají zajištěno kompatibilní chování. Větší část knihovny vychází ze „standardní knihovny V/V“ systému UNIX. Zde opět pro většinu programátorů nedochází k téměř žádným změnám.

Díky tomu, že jsou datové typy a řídicí struktury poskytované jazykem C podporovány přímo většinou počítačů, je knihovna nutná pro implementaci soběstačných programů velice malá. Funkce standardní knihovny jsou volány pouze explicitně, takže se jim lze vyhnout, nejsou-li potřeba. Většina z nich může být napsána v C a jsou přenositelné s výjimkou detailů operačního systému, které zakrývají.

I když C odpovídá schopnostem mnoha počítačů, je nezávislý na jakékoli konkrétní hardwarové architektuře. I s vynaložením malého úsilí je možné psát přenositelné programy, tedy programy, které mohou běžet bez úprav na různých hardwarových platformách. Stan-

dard jednoznačně hovoří o problémech s přenositelností a předepisuje seznam konstant charakterizujících počítač, na němž má program běžet.

C není silně typovaným jazykem, ale během jeho vývoje zesílila i jeho typová kontrola. Původní definice jazyka C sice nerada viděla záměnu ukazatelů a celých čísel, ale povolovala ji; to již déle neplatí a standard nyní požaduje správné deklarace a explicitní konverze, které již dříve vyžadovaly kvalitní kompilátory. Nové deklarace funkcí jsou dalším krokem tímto směrem. Kompilátory varují při většině typových chyb a neexistují automatické konverze nekompatibilních datových typů. Nicméně C si uchovává základní filozofii, že programátoři vědí, co dělají; pouze požaduje, aby své záměry uváděli explicitně.

C má stejně jako ostatní jazyky i své nedostatky. Některé operátory mají špatnou prioritu; občas by syntaxe mohla být lepší. Přesto se ukázalo, že C je nesmírně efektivní a expresivní jazyk, který našel uplatnění při vývoji širokého spektra aplikací.

Kniha je organizována následujícím způsobem. Kapitola 1 představuje kurz základů jazyka C. Smyslem této kapitoly je umožnit čtenáři co nejrychlejší start, protože jsme pevně přesvědčeni, že nejlépe se lze nový jazyk naučit psaním programů. Výuka předpokládá základní znalost programování; nevysvětlujeme zde pojmy, jako je počítač nebo kompilace (překlad programu) ani význam výrazu typu $n=n+1$. I když jsme se snažili předvádět užitečné techniky programování kde jen to bylo možné, kniha není koncipována jako referenční práce o datových strukturách a algoritmech; když jsme byli přinuceni volit, soustředili jsme se na jazyk.

Kapitoly 2 až 6 detailněji vysvětlují různé aspekty jazyka C formálněji než první kapitola, i když důraz je stále kladen na příklady kompletních programů, nikoli na izolované fragmenty. Kapitola 2 pojednává o základních typech dat, operátorech a výrazech. Kapitola 3 se zabývá řízením běhu programu: `if-else`, `switch`, `while`, `for` atd. Kapitola 4 probírá funkce a strukturu programu – externí proměnné, oblasti platnosti, práci s několika zdrojovými soubory atp., a také se zmiňuje o preprocesoru. V kapitole 5 jsou vysvětleny ukazatele a aritmetika ukazatelů. V kapitole 6 jsou vysvětleny struktury a unie.

Kapitola 7 popisuje standardní knihovnu, která poskytuje jednotné rozhraní operačního systému. Knihovnu definuje standard ANSI a měla by být podporována každým počítačem, který podporuje jazyk C, aby programy používající vstup a výstup mohly být přenášeny beze změny z jednoho systému na druhý.

Kapitola 8 popisuje rozhraní mezi programy v jazyce C a operačním systémem UNIX. Zaměřuje se na vstup a výstup, systém souborů a alokaci paměti. I přesto, že část této kapitoly je specifická pro operační systém UNIX, programátoři používající jiné systémy v ní naleznou užitečné informace včetně vzhledu do problematiky implementace jedné verze standardní knihovny a postřehy týkající se přenositelnosti.

Příloha A obsahuje referenční příručku jazyka. Oficiální definici syntaxe a sémantiky jazyka C představuje samotný standard ANSI. Nicméně tento dokument je určen především pro autory kompilátorů. Zdejší referenční příručka definuje jazyk stručněji, bez přehnaně formálního stylu. Příloha B je shrnutím funkcí standardní knihovny, opět spíše pro uživatele než pro implementátory. Příloha C je krátkým shrnutím změn oproti původnímu jazyku. Budete-li na pochybách, zůstává konečnou autoritou standard a váš kompilátor.