

Obsah

| | |
|---|-----------|
| Předmluva k aktuálnímu vydání | 16 |
| Předmluva k prvnímu vydání prvního dílu | 18 |
| 1 Úvod | 20 |
| 1.1 C – vznik, vývoj, charakteristika | 20 |
| 1.2 Typografické a syntaktické konvence | 21 |
| 1.3 Styl psaní programů | 22 |
| 2 Základní pojmy | 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 |
| 3 První začátky s C | 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 |

| | |
|---|----|
| 4 Terminálový vstup a výstup | 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 |
| 5 Řídící struktury | 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 |
| 6 Vstup ze souboru a výstup do souboru | 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>EOF</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 Vrácení přečteného znaku zpět do vstupního bufferu | 84 |

| | | |
|----------|--|-----|
| 7 | Typová konverze | 87 |
| 7.1 | Implicitní typová konverze | 87 |
| 7.2 | Explicitní typová konverze | 88 |
| 8 | Preprocesor jazyka C | 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 |
| 9 | Funkce a práce s pamětí | 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 const | 124 |
| | Modifikátor volatile | 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 .c souboru | 131 |
| | Doporučený obsah .h souboru | 132 |
| 9.5 | Inicializace jednoduchých proměnných | 140 |
| 10 | Pointery | 144 |
| 10.1 | Základy práce s pointery | 145 |
| 10.1.1 | Definice dat typu pointer 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 pointer NULL | 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 | Pointer na typ void | 154 |
| | Pointer na typ void jako pointer na několik různých typů | 154 |
| | Pointer na typ void 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 typedef | 159 |
| 10.5 | Pointerová aritmetika | 160 |
| 10.5.1 | Operátor sizeof | 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 navrácení 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 |
| 11 | Jednorozměrná pole | 173 |
| 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 |
| 12 | Řetězce | 192 |
| 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 |
| 13 | Vícerozměrná pole | 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 |
| 14 | Struktury, uniony a výčtové typy | 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 |

| | |
|--|-----|
| 15 Bitové operace a bitové pole | 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 |
| 16 Tabulka preferencí | 269 |