

Úvod

Aj keď o tom máloktorá učebnica programovania hovorí, jednou z najbežnejších činností, ktoré programátor robí, je štúdium cudzieho programu (a ak to nerobí, mal by to robiť). Učebnice sa obvykle sústredia na vysvetľovanie syntaxe a sémantiky daného jazyka, vymenujú jednotlivé príkazy a operátory a ich vlastnosti, uvedú postup pri riešení niektorých typických algoritmických problémov, a v najlepšom prípade uvádzajú zlomky programov, na ktorých je ukázaný konkrétny príklad použitia nejakej techniky.

Krátke úryvky programov však vyvolávajú ilúziu, že programy sú ľahko členiteľné na logicky uzavreté celky s dobre definovanými rozhraniami. Napokon, v učebniciach by aj bolo ťažké uvádzať rozsiahlejšie programy, s plnou zložitou vzťahov v rôznych úrovniach. A tieto vzťahy nie sú len navzájom medzi súčasťami programu, ale aj medzi programom a prekladačom, štandardnými knižnicami, operačným systémom a hardwarom. No a najmenej sa spomína jeden z najzložitejších vzťahov - vzťah medzi programom a ľuďmi, ktorí program píšu, udržiavajú, používajú, inštalujú, konfigurujú - či sú z neho frustrovaní a chcú by sa práce s ním čo najjednoduchšie a najrýchlejšie zbaviť.

Systematické vymenovanie jednotlivých súčastí jazyka je síce dôležité, ale na druhej strane je to určitým spôsobom „umelé“. Zdôrazňovanie formálnej stránky programovania neodráža „živú“ programátorskú prax, plnú chýb, omylov, pokusov a hľadania toho najvhodnejšieho prístupu. Učebnice obvykle nehovoria o rôznych tichých dohodách a zvyklostiach, tobôž o ich historickom pozadí, ktoré sa často stráca v temnotách programátorského praveku. A už vôbec nespomínajú takú nepatričnú vec, akou je kultúra a folklór programátorskej komunity.

Všetky uvedené javy sa najlepšie dajú ilustrovať na reálnom programe, alebo ešte lepšie na ucelenom knižničnom module - a to najmä na takom, ktorý už bol použitý vo viacerých programoch rôznymi programátormi, prípadne bol viackrát zmodifikovaný pre rôzne konkrétne potreby.

Uvádzanie cudzích programov by však mohlo narážať aj na problémy autorských práv, a tak obvykle útržky, ktoré sú uvedené, pochádzajú od jediného autora - priamo z pera autora učebnice. Čitateľ je tým pádom ochudobnený o kontakt s rôznymi druhmi konvencií, či už sa týkajú organizácie jednotlivých prvkov

programu, štandardov definícií a volaní, alebo formálnych záležitostí, ako je formátovanie a odsadzovanie zdrojového textu. Nevypracuje si určitú „imunitu“ či pochopenie pre variabilitu jazyka a pre rôzny kontext, v ktorom sú programy písané a používané.

Každý programátor však počas svojej praxe raz nevyhnutne príde do situácie, keď bude musieť cudzí program použiť, či upraviť, alebo opraviť. A ešte pravdepodobnejšie je, že niekto iný bude musieť použiť a upraviť jeho program. Preto sa musí naučiť cudzie programy nielen čítať a pochopiť, ale aj programy písať s ohľadom na existujúce konvencie a spôsobom, ktorý iným a aj sebe uľahčuje jeho pochopenie. A keďže tých vzťahov a súvislostí je veľké množstvo, sú v čase premenlivé a vzhľadom na ich široký záber je ťažké ich systematizovať, možno je najlepšou cestou tieto vzťahy ukázať a vysvetliť na skutočnom programe z reálneho sveta.

V nasledujúcich kapitolách budeme preto rozoberať práve takéto programy, riadok za riadkom. Dokonca sa nejedná ani o kompletný program: je to len jeden súbor z programu pozostávajúceho z viacerých súborov, aj keď práve v tomto súbore sa nachádzajú funkcie najvyššej úrovne vrátane funkcie `main()`. Jedná sa o súbor `stdiodemo.c` zo skupiny príkladov použitia knižnice `avr-libc` (štandardná knižnica pre použitie s prekladačom `avr-gcc` pre mikrokontroléry radu AVR¹ firmy Atmel). Adresa súboru je <http://www.nongnu.org/avr-libc/examples/stdiodemo/stdiodemo.c> a je okrem iného aj súčasťou inštalácie balíka WinAVR².

1 AVR je registrovaná ochranná známka firmy Microchip (predtým Atmel)

2 WinAVR je binárny balík GNU softwaru (okolo prekladača `avr-gcc`) pre vývoj programov pre mikrokontroléry Atmel AVR, určený pre OS Windows. Projekt je hostovaný na stránke <http://sourceforge.net/projects/winavr/>, v čase písania knihy bola posledná zverejnená verzia WinAVR20100110, ktorá je podľa vyjadrenia autora WinAVR Erica Weddingtona finálna. Aj keď obsahuje dnes už pomerne zastaralé verzie prekladača aj utilít, vďaka ucelenej suite doplnujúcich programov je stále dobrým základom pre prácu s mikrokontrolérmi AVR. Aktuálne binárne verzie samotného prekladača pod názvom AVR Toolchain ponúka firma Microchip (ktorá v roku 2016 odkúpila firmu Atmel) na adrese <https://www.microchip.com/mplab/avr-support/avr-and-arm-toolchains-c-compilers>.

Nebudeme sa však zaoberať samotnou funkčnosťou programu. Táto je napokon dostatočne podrobne vysvetlená v jeho nezvykle rozsiahlej a detailnej dokumentácii na adrese http://www.nongnu.org/avr-libc/user-manual/group_stdiodemo.html (a po inštalácii balíka WinAVR je dostupná aj na lokálnom disku ako [WinAVR]\doc\avr-libc\examples\stdiodemo), a v preklade aj v prílohe tejto knihy; čitateľ si ju môže preštudovať ešte skôr než začne čítať nasledujúce kapitoly. Jednotlivé riadky programu však použijeme ako námet pre úvahy súvisiace s prekladacím prostredím, použitím rôznych konštrukcií, ako aj spôsobom písania a čítania programov.

Program je zverejnený pod licenciou „BEER-WARE“, ktorá umožňuje aj takéto použitie programu. Nepochybne je tu tento program použitý užitočným spôsobom, a tak podľa licenčných podmienok som autorovi, ktorým je jeden z najaktívnejších vývojárov knižnice avr-libc a suity okolo avr-gcc, **Jörg Wunsch**, povinný zaplatiť pri našom eventuálnom stretnutí pivo, čo v takom prípade samozrejme učiním s tou najväčšou radosťou.

Poznámka k štandardu C99

Rovnako ako iné bežne používané programovacie jazyky, aj jazyk C je normalizovaný, konkrétne normou ISO/IEC 9899:1999³, ktorá je bežne označovaná ako C99. Predchádzajúce viac či menej formálne definované normy - K&R, (ANSI) C89/C90 - sa dnes uvádzajú len v historických súvislostiach.

Normovaný tvar jazyka dovoľuje na jednej strane programátorom písať programy, ktoré je možné prekladať rôznymi prekladačmi a na rôzne cieľové platformy s rovnakým správaním pri behu preloženého programu; na druhej strane to stanovuje jednoznačné pravidlá pre tvorbu takýchto prekladačov. Preto aj pri štúdiu jednotlivých javov a podrobností pri použití jazyka sa budeme v nasledujúcich kapitolách odvolávať na jednotlivé kapitoly tejto normy.

Samotné znenie normy je k dispozícii u organizácie ISO za príslušný poplatok. Na stránkach normalizačnej komisie je však k dispozícii aj voľne dostupná „draft“ verzia znenia normy, <http://www.open-std.org/jtc1/sc22/WG14/www/docs/n1256.pdf>, ktorá plne dostačuje pre všetky bežné potreby programátora.

3 http://www.iso.org/iso/catalogue_detail.htm?csnumber=29237

Poznámka k novším verziám štandardu (C11, C17/C18, C2x)

V roku 2011 bola prijatá nová verzia štandardu jazyka C, neformálne označovaná ako C11, oficiálne ISO/IEC9899:2011. Aj keď v nej boli prijaté niektoré novinky a samozrejme nastali aj niektoré menšie zmeny, opravy a úpravy, nie všetky prekladače a knižnice tieto zmeny podporujú či využívajú. Tieto zmeny zatiaľ ani nestihli preniknúť do každodennej programátorskej praxe. Z tohto dôvodu sú v texte knihy uvádzané odkazy primárne na štandard C99, pričom je pomerne jednoduché si relevantné state nájsť aj v novej podobe normy.

V roku 2018 bola prijatá verzia normy ISO/IEC 9899:2018, ktorá je označovaná obvykle ako C17. Jedná sa v nej len o opravy chýb v C11, bez nových vlastností, preto nemá príliš význam sa ňou bližšie zaoberať.

Nová verzia normy sa očakáva v roku 2023. Predbežne je označovaná ako C2x, aj keď už dnes sa používa aj označenie C23. Okrem iného odstraňuje relikty ako je K&R definícia funkcie a reprezentácie celých čísiel iné ako je dvojkový doplnok. Ak by táto kniha vznikla v dobe širokého rozšírenia tejto normy, aj tak by tieto relikty spomínala, práve kvôli ich historickým súvislostiam a vysvetleniu ich výskytu v reálne existujúcich programoch.

Poznámka k terminológii

Programovanie je pomerne nová oblasť ľudskej činnosti, ktorá sa naviac ešte stále búrlivo rozvíja. S tým súvisí problematika vytvárania výrazov, ktoré popisujú činnosti, predmety a javy, ktoré s programovaním súvisia. S týmto problémom sa vysporadúvajú z pochopiteľných dôvodov ako prví autori z anglofónnych krajín, častokrát využívajúc tradičnú flexibilitu anglického jazyka v preberaní cudzích výrazov, modifikovaní a menení významu existujúcich a jeho hravosť pri vytváraní úplne nových slov. Odborníci, ktorí chcú komunikovať v národných jazykoch, majú ťažšiu pozíciu: obvykle v ich jazyku neexistuje taká rozsiahla slovná zásoba pre činnosti, predmety a javy, hoci aj z iných, príbuzných odborov (ako napr. matematika), ktoré tiež používajú do značnej miery prevzatú terminológiu. Častokrát sú tiež tesne po vzniku určitej novinky súvisiace výrazy prirodzene priamo preberané z angličtiny, čo neskôr sťažuje ich nahrádzanie domácimi pojmami.

A tak je množstvo výrazov v súčasnej slovenskej či českej literatúre o programovaní jednoducho anglických, písaných pôvodným pravopisom aj vo viac či menej podarených fonetických prepisoch. Problematická je tým pádom aj ich gramatika - skloňovanie a časovanie. V tejto oblasti sa udomácnilo len niekoľko slovenských odborných výrazov - napríklad *smerník* (namiesto *pointer*), aj keď sa pre tento účel niekedy používa aj výraz *ukazovateľ* (česky *ukazatel*).

Napriek takémuto nedokonalému stavu sa nezdá, že by nejaký revolučný reformný krok pri tvorbe odbornej terminológie mal šancu na úspech. Jazyk, vrátane odborného, je viac predmetom postupnej evolúcie. Ani táto kniha si nedáva za úlohu stanovovať pravidlá a byť ukážkou vzorového použitia odbornej terminológie. Naopak, na mnohých miestach sa v nej vyskytujú výrazy, ktoré pravdepodobne nie sú aktuálne, správne či výstižné, za čo sa autor čitateľom touto cestou vopred ospravedlňuje.