Передмова

Дисципліна «Програмування мовою Сі» є базовою для багатьох курсів, що вивчаються студентамиза наступні чотири роки навчання. Базуючись на шкільних знаннях математики і фізики, вона в свою чергу служить фундаментом для вивчення інженерії апаратного та програмного забезпечення.

Під час вивчення курсу студенти знайомляться з основними принципами створення і візуалізації алгоритмів, парадигмою структурного програмування і реалізують її універсальною мовою програмування. Практичні навички, вироблювані курсом, містять алгоритмізацію і програмування інженерних завдань, аналіз, налагодження і тестування програмного коду.

Очевидно, що вибір першої мови програмування в значній мірі визначає стиль мислення початківця програміста, закладає основи підходів,в рамках яких надалі належить формувати навички та вміння, пов'язані з проблемно-орієнтованими мовами програмування, включно з високорівневими об'єктними підходами та моделями. Якою ж має бути перша мова програмування для вивчення майбутніми розробниками системного і прикладного програмного забезпечення?

Сі — це мова, створена свого часу для написання ОС Unix і доступна нині практично для будь-якої платформи, від мікроконтролерів до потужних обчислювальних кластерів.Сі на сьогоднішній день — основний засіб створення ядер операційних систем, драйверів пристроїв і просто високопродуктивного програмного коду. Крім того, синтаксис Сі, завдяки своїй структурованості і лаконічності ліг в основу безлічі високорівневих мов програмування, як загального призначення так і спеціальних: C++, C#, Java, PHP — лише кілька найбільш популярних прикладів.

Сі елегантна в синтаксисі, і дає змогу створювати дуже ефективні програми, які легко читаються і водночас легко переносяться на інші апаратні платформи. Однак ця мова створювалась для професійних системних програмістів, і тому є нетерпимою до неакуратного коду. Зворотною стороною високої ефективності є велика свобода, надана розробнику, на плечі якого лягає вся відповідальність за прийняті під час розробки програми рішення.

На відміну від мов, що спочатку створювалися як навчальні, Сі майже не контролює програміста. Рамки високорівневої мови програмування в Сі ілюзорні, їх легко обійти — через незнання або внаслідок прийнятих рішень. Концепції типів даних чітко окреслені, але при цьому зовсім прозорі, і крізь них проглядає апаратна архітектура, в якій все зводиться до машинних слів: послідовностей бітів фіксованої довжини, що задають дані або адреси комірок пам'яті. Сі має зручні та ефективні засоби роботи з векторними даними, такими як масиви і рядки, але самі абстракції цих типів зведені до режимів адресації, близьких за своєю суттю до реалізованих в апаратурі.

Сі довіряє програмісту, виконуючи будь які перетворення над типами або нестандартні поєднання синтаксичних конструкцій. Так можна створювати лаконічний і ефективний код— але так само легко зробити помилку, що приводить до катастрофічних результатів на етапі виконання програми. Перекладання відповідальності за дії програми на плечі програміста — крок неминучий у разі розробки системних програм, драйверів пристроїв і програм для високопродуктивних обчислень. Насправді цим і забезпечується швидкість написаного мовою Сі-коду.

Очевидно, що починати вивчати таку мову програмування краще на дуже простих програмах — наприклад таких, під час створення яких вивчаються основи алгоритмізації та структурного підходу. Практика показує, що програміст-початківець швидко засвоює як небезпечно допускати недбалість під час написання коду,виробляє необхідні навички і досить швидко починає створювати досить працездатний код.

Згодом, під час вивчення засобів створення складних програмних систем (наприклад, в рамках об'єктно-орієнтованого підходу) вироблена звичка до акуратного і вдумливого написання тексту програми повинна позитивно позначитися на якості створюваних програмних текстів і в підсумку привести до виходу на ринок праці висококваліфікованих інженерів-розробників електронно-інформаційних систем. Тому можна з упевненістю сказати, що мова Сі є прекрасним вибором для початкуючого системного програміста.