

Поняття алгоритму та програми

Поняття алгоритму

Інтуїтивно алгоритм визначається як “послідовність чітких недвозначних інструкцій, які зрозумілі виконавцеві і які призводять до певного результату за скінченний час”. Точне визначення алгоритму дати неможливо, але можна сформулювати ряд інтуїтивних вимог до алгоритмів.

Вважається, що послідовність інструкцій є алгоритмом, якщо вона задовольняє таким вимогам:

1. дискретність: алгоритм являє собою послідовність кроків, на кожному з яких виконується та чи інша інструкція; кожна наступна інструкція виконується після того, як завершиться виконання попередньої;
2. елементарність кроків: кожна інструкція є елементарною для виконавця і не вимагає від нього ніякої винахідливості;
3. локальність кроків: процес виконання інструкції не вимагає повернення до попередніх інструкцій або звертання до наступних;
4. детермінованість: після завершення чергового кроку завжди відомо, що робити на наступному кроці;
5. результативність: повинно бути визначено, що слід вважати результатом роботи алгоритму;
6. скінченність: результат повинен досягатися за скінченну кількість кроків;
7. масовість: алгоритм повинен бути призначений для вирішення не однієї конкретної задачі, а цілого класу однотипних задач.

Поняття програми

Програма, яка повинна виконуватися, записана в машинних кодах, тобто машинною мовою. Машинна мова передбачає запис інструкцій та операндів у вигляді послідовностей нулів та одиниць. Кажуть, що машинна мова є мовою низького рівня. Програма в машинних кодах являє собою послідовність команд, які процесор може сприйняти і виконати безпосередньо. Кожний тип машин має свою машинну мову. Машинний код, написаний для однієї машини, не може бути виконаний на іншій без використання спеціального програмного забезпечення.

Програмування в машинних кодах є дуже марудною справою. Програми є надто громіздкими і нерозбірливими. Важко тримати в пам'яті машинні коди і дуже легко припуститися помилки. Крім того, при переході на іншу машину все програмне забезпечення доводиться переписувати заново.

Але виявилось, що можна програмувати мовою, більш сприйнятною для людини, якщо автоматизувати процес перекладу з цієї мови на машинну. Ця ідея лягла в основу мов високого рівня.

Мови програмування високого рівня

Мови програмування високого рівня дозволяють писати програми в формі, більш наближеній до звичайної мови. Програму, написану мовою високого рівня, можна більш легко читати і модифікувати, і вони значно полегшують роботу програміста порівняно з написанням машинного коду.

Кожна мова високого рівня повинна мати формальний опис. У цьому описі повинні бути визначені правила запису програмних конструкцій (синтаксис) і те, яким чином ці конструкції виконуються (семантика). Транслятори пишуться у відповідності до цього формального опису. Формальний опис надає розробникам трансляторів певну свободу, і транслятори можуть бути реалізовані по-різному. Кажуть про те, що мова високого рівня може мати різні реалізації. Різні реалізації мови можуть мати відмінності в синтаксисі. Тоді може виникати ситуація, коли один транслятор сприймає програму нормально, а інший видає повідомлення про помилку.

Авторським варіантом мови називається опис мови у тому вигляді, у якому він був запропонований розробником (однією людиною або комітетом).

Написання програми

Компілятор сприймає програму, написану мовою високого рівня. Текст цієї програми називається початковим кодом. Результатом роботи компілятора є об'єктний код.

Різні частини програми можуть компілюватися незалежно. Програма, яка збирає в єдине ціле незалежно відкомпільовані об'єктні модулі, називається редактором зв'язків. При цьому відбувається налаштування програми на реальні машинні адреси, з якими вона буде працювати.

Результатом роботи редактора зв'язків є програма, готова до виконання (часто вона називається завантажувальним модулем). Ця програма може бути негайно виконана, а може бути збережена на диску.

Програма, яка завантажує готову програму в оперативну пам'ять і забезпечує її запуск на виконання, називається завантажувачем. Завантажувач у найбільш типовому випадку є частиною операційної системи. Він повинен розмістити програму в оперативну пам'ять і передати їй керування (це означає встановлення програмного лічильника на першу команду програми, що виконується).

Розрізняють 2 завантажувачі:

1. Абсолютний завантажувач розміщує програму за жорстко заданими адресами. Для цього в об'єктному і завантажувальному коді повинні бути визначені конкретні адреси комірок пам'яті, з якими буде працювати програма.
2. Налаштовувачий завантажувач важче реалізувати, але він є значно більш гнучким. Він може розмістити програму в довільному місці пам'яті. При цьому в завантажувальному коді вказуються не явні адреси, а зміщення відносно початку програми.

Дякую за увагу

Презентацію виконав:

Студент групи ПБ-13

Бабенко Ростислав