Лекция 7. Введение в программирование

Наш предмет трудно свести к какому-либо одному основанию. Так, необходимо общее представление о цифровой технике: регистрах, счетчиках, шинах. Необходимо знание родной для цифровой техники системы счисления – двоичной и ее связи с десятичной и шестнадцатеричной. Необходимо понимание архитектуры собственно процессора, важную роль в разработке которой сыграл Дж. Фон Нейман. Еще одним таким основанием микропроцессорной техники является дисциплина программирования.

История дисциплины программирования, т.е. разработки программ, как ни странно, начинается раньше, чем появились первые работающие программы. Соратником нашего знакомого профессора Бэббиджа в его, как мы теперь понимаем, безнадежном деле была Ада Лавлейс (урожденная Байрон, дочь поэта). В 1843 году она разработала первую программу для Аналитической машины Бэббиджа и заслуженно считается первым программистом. Её работы заложили теоретические основы программирования. В них впервые использован термин «цикл». В 80-е гг. 20 века разработан язык программирования Ада, названный в честь дочери поэта.

Первые работающие программы появились, конечно, сразу после появления первых компьютеров – МАРК-1, ЭНИАК. Первое время программирование осуществлялось в машинных кодах, т.е. в кодах непосредственно воспринимаемых процессора.

Морис Уилкс: ассемблер, библиотеки стандартных функций

Джон Бекус, IBM: фортран – циклы, массивы, запись в алгебраической форме. Предложил идею будущего языка в 1953. В 1957 г. был закончен компилятор Фортрана. Важнейший вывод – появилась возможность использования компьютеров не математиками, а пользователями в министерствах и крупных корпорациях

Основная тенденция – увеличения сферы использования компьютеров. Начали с военных и научных применений, перешли к коммерческим. Появляется специализация: Алгол 1958 (наука, инженерия, бизнес), Кобол 1959 (бизнес).

Постепенно усложнялись программы. Иногда затраты на программное обеспечение больших новых вычислительных систем доходили до 80% от их общей стоимости. Ко второй половине 60-х гг. на научных конференциях по вычислительным машинам признавали кризис программного обеспечения. Преодолеть сверхсложность программирования задач того времени позволила предложенная в 1968 г. Эдсгаром Дейкстрой парадигму структурного программирования.

Анализ проблемы, проведенный Дейкстрой показал, что проблема сверхсложности кода кроется в операторе безусловного перехода goto, который появился как калька команды ассемблера jump в языках высокого уровня. Программа, написанная с частым использованием оператора goto плохо воспринимается человеком. Как остроумно заметил один американский специалист по компьютерам, «если в вашей программе будет много операторов GOTO, то она станет похожа на спагетти».

Структурное программирование сначала было принято неохотно, однако первые успешные проекты, выполненные в структурном стиле, быстро убедили программистское сообщество в перспективности идей Дейкстры. В 1970-е годы парадигма структурного программирования стала мэйнстримом. Усложнение решаемых задач, однако, продолжалось. В 90-е годы от структурного программирования перешли к парадигме объектно-ориентированного программирования (ООП), которая доминирует и в настоящее время.

Однако, в относительно простых задачах, к которым относятся и задачи, решаемы с помощью микроконтроллеров, структурное программирование продолжает применятся. Язык программирования высокого уровня С появился в 1972 г. через три года после введения структурного программирования и конечно предполагает написание программ в духе этой парадигмы. Заметим, что язык С++, хотя и включает в себя язык С как подмножество, является объектно-ориентированным, поэтому эти языке правильнее считать дальними родственниками.

В нашем курсе мы программируем микроконтроллеры на языке С. Сначала рассмотрим подробнее, что собой представляет структурное программирование Дейкстры, а затем перейдем непосредственно к языку С.

Основные положения структурного программирования

I) Программа включает в себя следующие элементы

1) блок последовательных команд (императивное программирование)
2) ветвления
3) циклы

II) Вся программа делится на подпрограммы – часто повторяющиеся или просто логически отдельные участки кода

III) Разработка ведется сверху вниз. Нарисовать граф вызовов функций и объяснить последовательность.

Возможно, не совсем ясно, что же нового предложил Дейкстра. Для того, чтобы понять, что было до Дейкстры, рекомендуем ознакомиться с разделом сайта lurkmore.ru, посвященном индусскому коду и далее по ссылкам.

Парадигма структурного программирования математически обоснована теоремой Бёма-Якопини, согласно которой любой алгоритм можно представить в структурированном виде из блока последовательных команд, ветвлений и циклов.

Императивное программирование

Элемент микропроцессорной системы	Объект языка программирования
Ячейки памяти	Переменные
Команды перемещения из памяти в регистры и обратно	Обращение к переменным: запись (присваивание), чтение.
Условный переход	Управляющие структуры

Практически одновременно с императивным программированием (считая от разработки Фортрана) возникла другая ветвь языков программирования – декларативное программирование. В основном программирование у людей ассоциируется с одним из видов программирования – императивным (или процедурным). Процедурное программирование соответствует архитектуре вычислительных машин: программы на императивных языках программирования объясняют компьютеру *как* придти к желаемому вычислительному результату. В декларативных языках программирования объясняется *куда* надо придти, а компилятор языка формирует набор инструкций для достижения сформулированной программистом цели.

Пример: выборка из множества X подмножества A, такому, что элементы $a \in A$, удовлетворяют условию $\varphi(a) = 0$:

$$A = \{a \in X: \varphi(a) = 0\} \qquad (1)$$

На императивном языке это будет выглядеть так:

```
j = 1
for i = 1, size(X)
begin
   if j(X[i]) = 0
   begin
      A[j] = X[i]
      j = j + 1
   end
end
```

А на декларативном языке программа написана в формуле (1). Конечно, за исключением подробностей синтаксиса декларативного языка программирования. Одним из первых и поныне используемых языков программирования является Лисп, появившийся всего через год после Фортрана – в 1958 г.

Анекдот. Программа на декларативном языке очень высокого уровня

```
Хочу крутую базу данных!
```

Отладка программы:

```
Хочу крутую базу данных! Чтоб работало!
```