

И.А. Трашкова

Методические аспекты реализации циклических алгоритмов в локальной сети ЭВМ

Несмотря на то, что «белорусская программа базового курса информатики нацелена, прежде всего, на формирование пользовательской культуры учащихся» [1] в ней подчеркивается общекультурная значимость развития алгоритмического мышления школьников. Знакомство с учебным алгоритмическим языком начинается при изучении темы «Начала алгоритмизации» с объяснения базовых конструкций и принципов работы отдельных команд учебного алгоритмического языка. Наибольшие затруднения у школьников вызывает изучение команд повторения, таких как:

- цикл «n-раз»;
- цикл «для»;
- цикл «пока».

На наш взгляд, наглядным и дидактически обоснованным инструментом реализации циклических фрагментов алгоритмов является многопроцессорная ЭВМ. Однако, поскольку данный вид вычислительной техники недоступен сегодняшней школе, мы предлагаем преобразовать структуру программы не под архитектуру многопроцессорной ЭВМ, а под локальную сеть конкретного школьного кабинета ВТ с передачей части функций процессоров учащимся. Такое преобразование алгоритмов становится возможным благодаря разработанной нами методике сетевого взаимодействия школьников. Методика предполагает разделение задачи на более простые этапы, реализацию каждого этапа на отдельном рабочем месте ученика (РМУ) с последующей сборкой по сети полученных результатов. Для обеспечения наглядности строится диаграмма этапной зависимости – блочная структура, которая отражает номер задействованной РМУ и последовательность передачи вычислительного процесса. Наиболее сложно реализуемые функции, такие как инициирование вычислительного процесса и взаимодействие процессоров, при этом выполняются учащимися. Основным условием достижения понимания учеником принципов работы многопроцессорной ЭВМ становится исполнение им ролевой функции одного из процессоров с обязательной последующей рефлексией своей деятельности. Здесь ситуация подобна той, которая возникает при изучении темы «Начала алгоритмизации» на игровых задачах: ученик легко объясняет «на словах» алгоритм решения, в то время как запись алгоритма на алгоритмическом языке представляет проблему. В случае параллельных алгоритмов это расхождение между «словом и делом» становится еще более сильным. Предложенная реализация циклических алгоритмов способствует, с одной стороны, развитию алгоритмического мышления за счет модификации способа построения алгоритма, с другой – пониманию работы многопроцессорной ЭВМ за счет выполнения коллективом учеников функций процессора.

Рассмотрим возможные, с алгоритмической точки зрения, способы реализации в локальной сети алгоритмов с циклическими участками:

- параллельный;
- последовательный;
- параллельно-последовательный.

Работа в стиле параллельных ЭВМ может быть разбита на следующие этапы:

1. на рабочее место ученика (РМУ) передаются объекты обработки: рабочие файлы и исходные данные;
2. выделенные этапы выполняются на всех РМУ;
3. готовые решения собираются в единый проект.

Количество задействованных РМУ в этом случае определяется изначально по формуле:

$$[(X_k - X_n)/N] + 1$$

где X_k – конечное значение параметра цикла, X_n – начальное значение параметра цикла, N – шаг изменения параметра цикла, $[X]$ – целая часть X .

Работа в стиле последовательных ЭВМ может проводиться следующим образом:

1. объект обработки (рабочий файл и исходные данные) передается на стартовое РМУ;
2. проверяется условие выхода из цикла;
3. выполняется выделенный этап за конкретным РМУ;
4. рабочий файл передается на следующее РМУ и осуществляется переход ко 2-му пункту.

Число задействованных РМУ определяется динамически во время выполнения алгоритма.

Определим критерии выбора способа реализации конкретного циклического алгоритма. Распараллеливание алгоритма желательно только в случае, когда алгоритм можно реализовать с помощью цикла «для» с произвольным порядком присвоения значений параметру цикла, при этом нас не интересует какие конкретные действия выполняются в теле цикла. В случае, если циклический алгоритм реализован с циклом «пока», то такой алгоритм предполагает его последовательное исполнение в локальной сети.

Наиболее интересным, на наш взгляд, является распараллеливание циклических участков алгоритмов, т.е. преобразование последовательной программы в параллельную для реализации ее на многопроцессорной ЭВМ. «Распараллеливанием называют комплексную процедуру «подгонки» алгоритма, программы или даже вычислительного метода к архитектуре конкретного или даже гипотетического комплекса» [2].

Распараллеливание алгоритма в локальной сети представляет собой один из возможных способов развития алгоритмического мышления школьников. С одной стороны, это становится возможным за счет особенностей самой методики, таких как разделение общей задачи на более простые части, упрощение реализации отдельных этапов, неоднократное моделирование алгоритма, преобразования исходных данных и анализ правильности как окончательного, так и промежуточных результатов. С другой стороны, за счет составления и реализации параллельных, новых для школьного курса информатики, алгоритмов. Использование компьютерной коммуникации в структуре урока позволяет сместить акцент с программирования на алгоритмизацию и рассматривать локальные сети как средство, повышающее эффективность развития алгоритмического мышления школьников.

В качестве примера рассмотрим задачу «О банковском проценте». «В сберегательный банк в начале года положена сумма в 100 000 р. На вклад ежегодно начисляется 30%. Вопросы:

1. Когда сумма вклада достигнет 300 000 р.?
2. Какой будет сумма вклада после 10 лет ? [3].

В случае, когда предполагается простой процент начисления, т.е. процент каждый год начисляется на основной вклад, появляется возможность распараллелить алгоритм. При этом тело цикла представляет собой автономную структуру, которая выполняется на каждом РМУ с определенным значением параметра. Последовательность выполнения действий за отдельными РМУ не имеет значения, т.к. осуществляется информационное взаимодействие школьников над общей памятью с асинхронной обработкой исходных данных. Главным и единственным требованием является соответствие числа задействованных РМУ количеству лет, которые вклад пролежал на данном счете. Диаграмма этапной зависимости в этом случае будет отражать не только передачу управления между отдельными РМУ, но и потактовое распределение работы.

В случае, когда предполагается капитализированный процент начисления, т.е. каждый год процент начисляется на новую сумму вклада, распараллеливание алгоритма нецелесообразно, т.к. циклический алгоритм реализован с использованием рекуррентных соотношений, а действие за каждым последующим РМУ предполагает использование результатов, полученных на предыдущем этапе. Такой алгоритм может быть разрешен последовательной сборкой частей в локальной сети.

Итак, при внешнем усложнении, основанном на включении средств коммуникации и превращении класса в многопроцессорную ЭВМ, возможен эффект упрощения для школьника материала информатики, основанный на разделении структур данных и программ на исходные элементы: простые элементы и команды.

ЛИТЕРАТУРА

1. **Быкадоров Ю.А., Кузнецов А.Т.** Школьная информатика в Республике Беларусь // Информатика и образование, 1999. Вып. 3. С. 28–32.
2. **Вальковский В.А.** Распараллеливание алгоритмов и программ. Структурный подход. М.: Радио и связь, 1989. – 176 с.
3. **Быкадоров Ю.А., Кузнецов А.Т.** Информатика: Учеб. пособие для 9-го кл. общеобразоват. шк. Мн.: Нар. асвета, 1995. – 253 с.

S U M M A R Y

The problems of realizing the cyclic algorithms in the local net at the informatic lessons in the 8-9 forms of the basic school are considered. It contributes to the development of the algorithmic thinking of the pupils thanks to modification of the way of algorithm making. The work in the style of successive and parallel computers are considered.