

отсутствие «изолированных»; повышение учебной мотивации учащихся; улучшение детско-родительских отношений; улучшение взаимодействия в системе учитель-ученик; самоопределение дальнейшего образовательного маршрута (учащиеся 9-11 классов); повышение социальной и коммуникативной компетентности учащихся; включение учащихся в активную школьную и внеурочную деятельность.

УДК 621.762.4

Шутько Е.И.

ОСНОВНОЙ АЛГОРИТМ РАБОТЫ ПРОЦЕССОРА

БГПУ, Минск

Все гениальное – просто. Гениальное – это компьютер. Просто – это нуль и единица.

«Компьютер – это совокупность программных и аппаратных средств» – ни раз повторяемая учителем информатики фраза. Вся информация – числа, символы, звук, графика – переводится в двоичную систему – это тоже школьнику известно. Но связи нуликов и единичек с устройствами компьютера они не улавливают. Эти темы в школьном курсе информатики существуют отдельно друг от друга.

Устройства компьютера и программы тоже не связаны между собой, пока компьютер выключен. Как только компьютер включается, приходит в действие основной алгоритм работы процессора, аппаратные и программные части становятся единым целым. Учитывая важность понимания сути работы процессора, а значит, и компьютера в целом без темы «Основной алгоритм процессора» раздел «Основы компьютера» является не законченным.

Для углубленного изучения архитектуры компьютера предлагается разработка лекции «Реализация базовых структур в ассемблере. Пошаговое выполнение программ по основному алгоритму работы процессора». Предполагается, что при изучении данной темы ученики знают логическое

распределение памяти, сегментную адресацию, структуру программы на ассемблере и умеют создавать *.exe и *.com файлы.

Тема лекции «Реализация базовых структур в ассемблере»

1 Написание программ с ветвлениями

1.1 Условные переходы Jnnn (больше, меньше, равно)

Пример 1. Составить программу поиска максимального из двух чисел, находящихся в ячейках памяти X и Y. Результат поместить в Z.

Алгоритм решения этой задачи опишем в виде программы на языке Pascal. Текст программы и результат ее выполнения представлен на рисунке 1.

Далее запишем соответствующую программу с использованием команд процессора (таблица 1). Условно будем считать, что разместилась программа в памяти компьютера с адреса 1500, и все команды занимают по два байта. Под адресом, для простоты, будем понимать смещение.

<pre>[] begin readln(x, y); z:=y; if x>y then z:=x; writeln('z=', z); Readln end.</pre>	<pre>MODEL SMALL STACK 256 DATASEG X dw 6 Y dw 7 Z dw ? CODESEG Begin: mov ax, @data mov ds, ax mov ax, [X] cmp ax, [Y] jge short M1 mov ax, [Y] M1: mov [Z], ax mov ah, 4Ch mov al, 00h int 21h End begin</pre>
<pre>? 6 z=?</pre>	

Рисунок 1

Несокращенный вид команды ветвления if, используемый в языке программирования Pascal, реализован в ассемблере с помощью команды условного перехода Jge (больше или равно).

Таблица 1

...	
Переслать хв ах	1500
Сравнить ах и у	1502
Если больше или равно, переход на +1 слово	1504
Переслать у в ах	1506
Переслать ах в z	1508
Стоп	1510
...	

Никакого "if" в ассемблере нет, и эту операцию приходится осуществлять в два этапа. Команда `cmp` позволяет сравнить два числа, сохраняя результат своей работы во флагах. Флаги – это биты специального регистра. Достаточно запомнить три основных состояния: меньше (`bellow` или `less`), больше (`above` или `great`) и равно (`equal`). Семейство команд условного перехода `Jxx` проверяют условие `xx` и, если оно истинно, совершают прыжок по указанному адресу. Например, `JE` прыгает, если числа равны (`JumpifEqual`), а `JNE`, если неравны (`JumpifNotEqual`). `JB/JA` работают с беззнаковыми числами, а с `JL/JG` – со знаковыми. Любые два не противоречащих друг другу условия могут быть скомбинированы друг с другом, например, `JBE` – переход, если одно беззнаковое число меньше другого или равно ему.

Безусловные переходы `Jmp`

Безусловный переход осуществляется командой `Jmp`. Применение этой команды, например, необходимо при реализации на ассемблере несокращенного варианта команды ветвления.

Структурный вид этой команды на языке `Pascal` и в машинных командах изображен на рисунке 2).

Архитектура современного компьютера, как и 70 лет назад строится на основе фон-неймановский принципов, основными из которых являются следующие: двоичное представление информации; каждая ячейка памяти имеет адрес; программное управление.

```

program max_x_y;
var x,y,z: integer;
begin
readln(x,y);
if x>y then z:=x
  else z:=y;
writeln('z=',z);_
readln
end.

```

```

Turbo Pa
6
6
?
z=?

```

```

MODEL SMALL
STACK 256
DATA SEG
X dw 6
Y dw 7
Z dw ?
CODE SEG
Begin:
    mov ax, @data
    mov ds, ax

    mov ax, [X]
    mov bx, [Y]
    cmp ax, bx
    Jshort M1
    Mov dx, bx
    jmp m2
M1:
    Mov dx, bx
M2:
    Mov [Z], dx
    mov ah, 04Ch
    mov al, 0h
    int 21h
End begin

```

Рисунок 2

Разработанный выше учебный материал является наглядной и убедительной демонстрацией пока незыблемых архитектурных принципов компьютера.

УДК 378.147.31 (476.2)

Юрис С.А., Юрис Т.А.

**О ВИЗУАЛИЗАЦИИ УЧЕБНОГО МАТЕРИАЛА
ПО СОЦИАЛЬНО-ГУМАНИТАРНЫМ
ДИСЦИПЛИНАМ**

ГГТУ, БТЭУ ПК, Гомель

Значительная часть профессорско-преподавательского состава учреждений высшего образования принадлежит к поколению X (принятое в психологии обозначение для родившихся до 1983 года), а это значит, сформировалась личностно и профессионально в ту эпоху, когда основным средством передачи информации было слово. Нынешнее студенчество же – поколение Y – продукт иной эпохи. В последние десятилетия мы вошли в оптическую цивилизацию, цивилизацию зрения, в которой культура распространяется, в первую очередь, с помощью картинки. Литература как основная фабрика по выработке культурных значений отступила под натиском сложных