Информатика. Упражнение 6
Три уровня языков программирования

Цель работы: познакомиться с различиями в программирования в машинных кодах, на языке ассемблера и на языке высокого уровня.

Чаще всего программы пишутся на языках высокого уровня, реже - на языках ассемблера и в исключительных случаях (обычно для управляющих ЭВМ) - в машинных кодах, но на ЭВМ выполняются только программы в машинных кодах. Перед выполнением программа должна быть транслирована (переведена) с языка высокого уровня или с языка ассемблера в машинные коды. Необходимым условием высокой квалификации программиста является понимание принципов выполнения программы на ЭВМ.

Задание

1. Изучите следующие разделы описания упражнения 6:
   - структура ЭВМ;
   - система команд ЭВМ;
   - форматы чисел;
   - язык ассемблера;
   - язык высокого уровня.
   - Пример программы на трёх языках.
2. Возьмите у преподавателя номер задачи.
3. Запрограммируйте задачу на всех трёх языках

Структура ЭВМ

Рассмотрим простейшую гипотетическую ЭВМ, структура которой изображенна на рис. 1. ЭВМ состоит из процессора, оперативной памяти, устройсва ввода (клавиатуры) и устройсва вывода (дисплея).

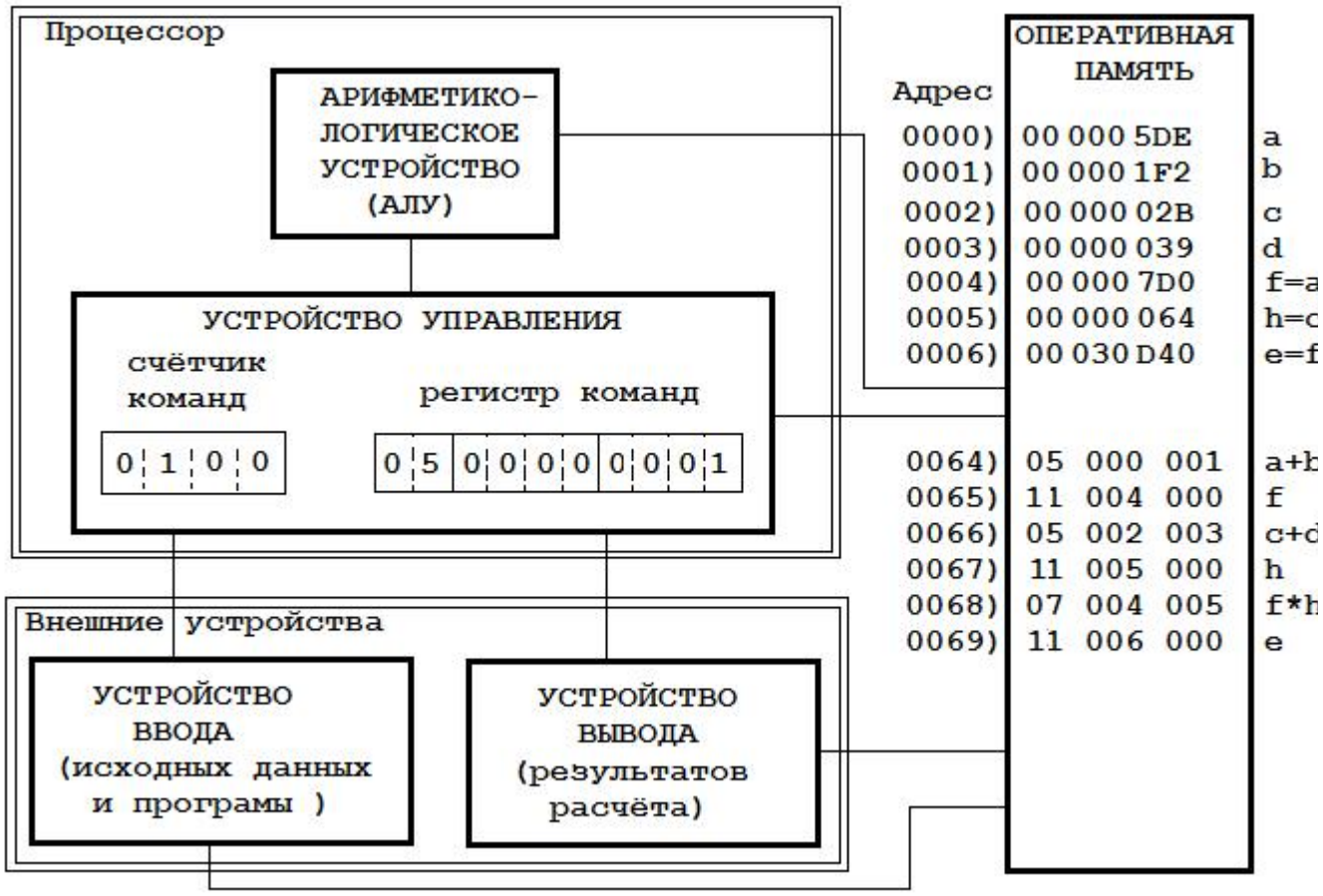


Рис. 1. Структура ЭВМ

Все числа на рисунке - в 16-й системе счисления.

Система команд

В отличие от большинства современных ЭВМ адреса в нашей ЭВМ имеют не байты, а 36-разрядные ячейки памяти. Команды ЭВМ имеют следующую структуру:

Адреса в команде - 15-разрядные. Максимальный адрес - 2^{15}-1 = $7FFF_{16}$ = 32767_{10}. Максимальное количество команд - $2^6 = 64_{10}$. Для выполнения упражнения понадобятся следующие 16 команд.

Арифметические операции

Результат арифметической операции остаётся в АЛУ на сумматоре.

0 1	A1	A2

- сложение с плавающей запятой операндов с адресами A1 и A2

02	A1	A2

- вычитание с плавающей запятой из операнда с адресом A1 операнда с адресом A2

03	A1	A2

- умножение с плавающей запятой операндов с адресами A1 и A2

04	A1	A2

- деление с плавающей запятой операнда с адресами A1 на операнд с адресом A2

05	A1	A2

- сложение с фиксированной запятой операндов с адресами A1 и A2

06	A1	A2

- вычитание с фиксированной запятой из операнда с адресом A1 операнда с адресом A2

07	A1	A2

- умножение с фиксированной запятой операндов с адресами A1 и A2

08	A1	A2

- деление с фиксированной запятой операнда с адресами A1 на операнд с адресом A2;
остаток от деления отбрасывается

Операции пересылки

10	A1	A2

- пересылка из ячейки с адресом A1 в ячейку с адресом A2

11	A1	

- пересылка из сумматора в ячейку с адресом A1.

Операции перехода

12	A1	

- безусловный переход в ячейку с адресом A1.

13	A1	A2

- условный переход. Если содержимое сумматора <0, то - по адресу A1, иначе переход по адресу A2.

Операции ввода-вывода

20	A1	

- ввод с клавиатуры в ячейку с адресом A1 целого числа.

21	A1	

- ввод с клавиатуры в ячейку с адресом A1 числа с плавающей запятой.

30	A1	

- вывод на экран дисплея из ячейки с адресом A1 целого числа.

31	A1	

- вывод на экран дисплея из ячейки с адресом A1 числа с плавающей запятой.

Операции завершения программы

0		

- останов. В результате выполнения этой команды счётчик команд не меняется. Точнее, в конце выполнения команды *Останов* в счётчике восстанавливается её адрес.

В реальной ЭВМ алгоритмы ввода и ввода реализуются с помощью сложных аппаратных средств и специального программного обеспечения (драйверов и подсистемы ввода-вывода, входящей в состав операционной системы). Для упрощения выполнения упражнения будем считать, что ввод или вывод содержимого ячейки делается одной машинной командой. Пользователь может вводить с клавиатуры только шестнадцатиричные цифры, т.е. символы 1, 2, . . , 10, A, B, . . ,F, знак *минус* и точку. На экран дисплея содержимое ячейки памяти выводится в шестнадцатиричной системе счисления.

На рис. 1 в оперативной памяти записаны программа и данные для вычисления выражения:

e = (a + b) * (c + d),
где a=1502, b=498, c=53, d=47 в дес. системе.

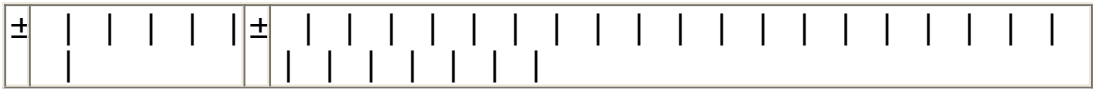
Предполагается, что значения a, b, c и d предворительно записаны в память.

Форматы чисел

Формат числа с фиксированной запятой (целого)

Максимальное целое число равно 2^{35}-1 = 34 359 738 367_{10} = 7FF FFF FFF_{16} .

Формат числа с плавающей запятой



В разряды 0 - 7 записывается порядок числа. В разряде 0 - знак порядка.

В разряды с 8-го по 35-й помещается мантисса. В 8-м разряде - знак мантиссы.

Язык ассемблера

Программирование в машинных кодах очень трудоёмко. Программа в кодах трудно читается, в ней легко сделать ошибку. Кроме того, возникает задача распределения памяти, которая на языках более высокого уровня выполняется транслятором. Программа на языке ассемблера - это тоже последовательность машинных команд, только коды команд заменены сокращёнными названиями, а адреса - идентификаторами переменных. Например, команда сложения целочисленных переменных a и b на языке ассемблера записывается так:

```
AddC a b
```

В табл.1 дан список команд ассемблера, соответствующих описанным выше машинным кодам.

Табл. 1. Команды ассемблера

№ п. п.	Имя команды	Операнды	Описание команды	Пример
Арифметические				
1	AddP	Операнд1 Операнд2	Сложение чисел с пл. запятой. Результат - на сумматоре	AddP a b
2	DedP	Операнд1 Операнд2	Вычитание чисел с пл. запятой. Результат - на сумматоре	DedP a b
3	MulP	Операнд1 Операнд2	Умножение чисел с пл. запятой. Результат - на сумматоре	MulP a b
4	DivP	Операнд1 Операнд2	Деление чисел с пл. запятой. Результат - на сумматоре	DivP a b
5	AddC	Операнд1 Операнд2	Сложение целых чисел. Результат - на сумматоре	AddC a b
6	DedC	Операнд1 Операнд2	Вычитание целых чисел. Результат - на сумматоре	DedC a b
7	MulC	Операнд1 Операнд2	Умножение целых чисел. Результат - на сумматоре	MulC a b
8	DivC	Операнд1 Операнд2	Деление целых чисел. Результат - на сумматоре	DivC a b
Команды пересылки				
9	Send	Операнд1 Операнд2	Пересылка: Операнд1 → Операнд2 Эту команду можно использовать для присвоения переменной значения константы	Send a b Send 5 a
10	SendS	Операнд1	Пересылка: Сумматор → Операнд1	SendS a
Команды перехода				
11	Go	Метка1	Безусловный переход к команде с меткой	Go M1 . . . M1: AddC a b
12	Golf	Метка1 Метка2	Условный переход. Если содержимое сумматора <0,	Golf M1 M2

			то - Метка1, иначе Метка2.	M1: AddC a b . . . M2: AddC d e
Команды ввода-вывода				
13	RC	Операнд1	Ввод с клавиатуры в Операнд1 целого числа.	RC a
14	RP	Операнд1	Ввод с клавиатуры в Операнд1 числа с плавающей запятой.	RP a
15	WrC	Операнд1	Вывод на экран дисплея Операнда1 целого типа.	WrC a
16	WrP	Операнд1	Вывод на экран дисплея Операнда1 с плавающей запятой.	WrP a

Язык высокого уровня

Для выполнения упражнения понадобится кроме машинных команд и языка ассемблера упрощённый язык высокого уровня. Опишем его.

В описании языка будем использовать два обозначения

::= - "это есть";
| - "или".
Пример
знак числа ::= + | -
Читается так: *знак числа это есть плюс или минус.*

Алфавит: латинские буквы, арабские цифры.

Комментарий начинается символами // и распространяется до конца строки.

Константы - десятичные числа.

Идентификаторы переменных. Все переменные делятся на скаляры и массивы. Идентификатор переменной начинается с буквы и состоит из букв и цифр.

Элемент массива обозначается так:

Идентификатор_переменной[номер элемента]
Примеры:
Ar1[5]
E[0]

Идентификатор_переменной[номер элемента] будем называть идентификатором элемента массива.

Операции
Арифметические: +, -, *, /.
Сравнения: ==, >, <, <=, >=.

Операторы языка

Операторы объявления переменных

Целого типа
int идентификатор_скаляра;
int идентификатор_массива(длина массива);
Вещественного типа
real идентификатор скаляра;
real идентификатор массива(длина массива);

Примеры

```
int a; //скаляр a целого типа
real Ar(10); //массив Ar из десяти элементов
```

Оператор присваивания

Идентификатор_скаляра | Идентификатор элемента массива = арифметическое выражение;
Арифметическое выражение может состоять из одного идентификатора переменной.
Примеры

```
a = Ar[2];
f = (a + b) * (c + d);
F[4] = Ar[3];
```

Оператор условия

```
if(условие)
   блок операторов
```

или

```
if(условие)
   блок операторов
else
   блок операторов
```

Блок операторов ::= оператор | последовательность операторов, заключённая в фигурные скобки.
Условие ::= арифметическое_выражение значок_операции_сравнения арифметическое_выражение
Пример

```
if(a > b)
  c = 1;
else
{  c = 2;
  d = c +e;
}
```

Операторы цикла

Оператор *while* - делать, пока выполняется условие

```
while(условие)
  блок операторов
```

Пример

```
//Подсчитывается сумма арифметической прогрессии
N = 1;
s = 0;
while(N <= 10)
```

```
{  s = s + N;
   N = N + 1;
}
// s = 55
```

Оператор *for* - делать, заданное число раз.

```
 for(начальное значение переменной цикла; условие продолжения цикла; изменение переменной цикла)
    блок операторов
Пример
//Подсчитывается сумма арифметической прогрессии
s = 0;
for( i=1; i <= 10; i = i +1 )
{ s = s + 1;
}
// s = 55
```

Операторы ввода-вывода

Оператор *read* - ввод с клавиатуры

```
   read Идентификатор_скаляра | Идентификатор элемента массива;
Примеры
read a;
read Ar[2];
```

Оператор *write* - вывод на экран

```
   write Идентификатор_скаляра | Идентификатор элемента массива;
Примеры
write a;
write Ar[2];
```

Пример программы на трёх языках

Задача. Вводятся 10 натуральных чисел. Найти среди них наибольшее и наименьшее

Решение в машиных кодах

Адрес ячейки памяти	Данные или команда	Комментарии
000)	9	K - количество вводимых чисел минус 1
001)	0	N_{min}
002)	0	N_{max}
003)		N_i - текущее введённое число
004)	1	шаг цикла
. . . .	. . . .	
. . . .	. . . .	
. . . .	. . . .	

008)	20	001	000	Ввод начального значения N_{min}
009)	10	001	002	Начальное значение $N_{max} = N_{min}$
00A)	20	003	000	Ввод N_i
00B)	06	003	001	$N_i - N_{min}$
00C)	13	011	00D	Если $N_i < N_{min}$, то перейти по адресу *011*
00D)	06	003	002	$N_i - N_{max}$
00E)	13	012	00F	Если $N_i > N_{max}$, то перейти по адресу *00F*
00F)	10	003	002	$N_{max} = N_i$
010)	12	012	000	Перейти в 012)
011)	10	003	001	$N_{min} = N_i$
012)	06	000	004	АЛУ = K-1
013)	10	000	000	K - АЛУ - уменьшить счётчик
014)	13	015	00A	Если K >= 0, то перейти по адресу *00A*
015)	30	001	000	Вывод на экран N_{min}
016)	30	002	000	Вывод на экран N_{max}
016)	00	000	000	Останов
. . . . .	. . . . .			
. . . . .	. . . . .			
. . . . .	. . . . .			

Решение на языке ассемблера

```
   Send 9 K              //K = 9     количество вводимых чисел минус 1
   RC Nmin               //Ввод начального значения Nmin
   Send Nmin Nmax        //Начального значения Nmax = Nmin
   Send 1 i              //i = 1     шаг цикла
M5: RC   N               //Ввод N
   DedC N Nmin           //N - Nmin
   Golf M1 M2            //Если N < Nmin, то перейти к M1
M2: DedC N Nmax          //N - Nmax
   Golf M3 M6            //Если N <= Nmax, то перейти к M3
M6: Send N Nmax          //Nmax = N
   Go   M3
M1: Send N Nmin          //Nmin = N
M3: DedC K i             // Сумматор = K - i
   SendS K               //K = Сумматор
   Golf M4 M5            //Если K >= 0, то перейти к M5
M4: WrC Nmin
   WrC Nmax              // Вывод Nmax
```

Решение на языке высокого уровня

```
int Nmax;
int Nmin;
int N;
int i;
read Nmax;
```

```
Nmin = Nmax;
for (i = 1;i<=10;i = i + 1)
{ read N;
  if(N < Nmin) Nmin = N;
          else if (N > Nmax) Nmax = N;
}
write Nmin;
write Nmax;
```

Задачи

Задача 1

Ввести N натуральных чисел. Подсчитать, сколько среди введённых чисел чётных и сколько нечётных.

Задача 2

Ввести натуральное число. Подсчитать содержащееся в нём количество простых множителей, равных пяти.

Задача 3

Ввести 2 натуральных числа: A и B. Если A делится на B, то вывести частное, иначе вывести оба числа.

Задача 4

Ввести натуральное число N. Подсчитать N!. Если N! не вмещается в ячейку памяти (переполнение), вывести -1.

Задача 5

Вычислить приближённое значение натурального числа e по формуле:

```
e = 1 + 1/1! + 1/2! +1/3! + 1/4!;
```

Задача 6

Ввести 2 натуральных числа: A и B. Вычислить $C = A^B$. Если C не вмещается в ячейку памяти (переполнение), вывести -1.

Задача 7

Вычислить по формуле Герона корень введенного целого числа A.

Алгоритм Герона для вычисления квадратного корня числа a:
Ввести положительное число a.
Принять в первом приближении x1 = a/2;
Вычислять по формуле xi = (a/xi-1 + xi-1)/2, пока |a - xi*xi| > 0.001*a.

Задача 8

Ввести три числа и вывести их в порядке возрастания.

Задача 9

Ввести в произвольном порядке три положительных и три отрицательных числа. Вывести сначала отрицательные числа, а затем - положительные.

Задача 10

Ввести натуральное число *a*. Определить, в какой из следующих диапазонов оно попадает:

1. $0 \le a \le 10$;
2. $10 < a \le 100$;
3. $100 < a \le 1000$;
4. $a > 1000$.

Вывести номер диапазона.

Задача 11

Ввести N чисел $n_1, n_2, \ldots, n_N$, $N > 3$.
Вывести n_i, если $n_i > n_{i-1}$, i =2, 3, .. ,N.

Задача 12

Найти число Фибоначчи с номером n (n>1). Каждый член последовательности Фибоначчи является суммой двух предыдущих $x_n=x_{n-1}+x_{n-2}$, $x_0=0$, $x_1=1$.

Задача 13

Ввести N чисел. Найти их произведение. Если результат не умещается в ячейку памяти (переполнение), вывести -1.

Задача 14

С клавиатуры вводится последовательность из N чисел. Определить, сколько из них больше пяти.

Задача 15

Даны два натуральных числа X и Y. Составить программу для вычисления суммы кубов всех четных чисел, лежащих в диапазоне [X, Y]

Задача 16

Ввести коды ASCII N символов. Выбрать из них и вывести только коды цифр.

Задача 17

Вводить натуральные числа до тех пор, пока не выполнится условие:

$$N_i \geq N_{i-1} + N_{i-2}.$$

Задача 18

С клавиатуры вводится последовательность из N чисел, найти максимальное из них.

Задача 19

Вводить натуральные числа до тех пор, пока не будут введены три таких числа *a, b* и *c*, что

$$a^2 + b^2 = c^2.$$

Выведите эти числа.

Задача 20

Вводить натуральные числа до тех пор, пока не будет введено такое число *a*, что

$$a + a = a * a = a^2. \quad (1)$$

Условие (1) должно проверяться в программе, даже если Вы догадались, что это за число.

Задача 21

Определить, является ли сумма N введенных натуральных чисел четной.

Задача 22

Дано целое m >1. Получить наибольшее целое k, при котором 4^k<m.

Задача 23

Найти сумму целых положительных чисел, кратных четырем, из промежутка [A, B]. Значения A и B вводятся с клавиатуры.

Задача 24

Определить, является ли данное натуральное число N факториалом какого-нибудь числа, если «да», то какого.

Задача 25

Вывести все двузначные числа, у которых первая цифра больше второй.

Задача 26

С клавиатуры вводится последовательность из N чисел. Определить, сколько из них больше нуля и меньше N.

Задача 27

Найти первое число Фибоначчи, большее заданного натурального n (n>1). Каждый член последовательности Фибоначчи является суммой двух предыдущих $x_n=x_{n-1}+x_{n-2}$, $x_0=0$, $x_1=1$.

Задача 28

Ввести N чисел. Найти среднее арифметическое положительных из них.

Задача 29

Дано число x. Вычислить $1 + 1/x + 1/x^2 + 1/x^3 + 1/x^4 + 1/x^5$.

Задача 30

Ввести N чисел. Найти произведение тех, которые по модулю не превышают 1 и не равны 0.