**Министерство науки и высшего образования Российской Федерации**

федеральное государственное автономное образовательное учреждение

высшего образования

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

**Отчет**

по лабораторной работе № 4

по дисциплине «**Основы профессиональной деятельности**»

Вариант №41333

Автор: Дениченко Александр Олегович

Факультет: ПИиКТ

Группа: P3112

Преподаватель: Осипов Святослав Владимирович

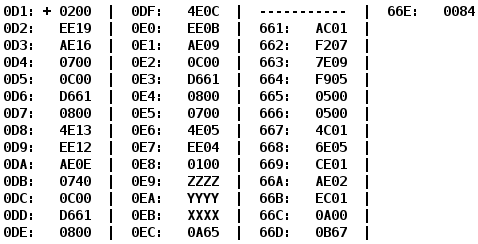


Санкт-Петербург, 2023

**Цель** **работы:**

По выданному преподавателем варианту восстановить текст заданного варианта программы и подпрограммы (программного комплекса), определить предназначение и составить его описание, определить область представления и область допустимых значений исходных данных и результата, выполнить трассировку программного комплекса.

**Задание:**



**Ход работы:**

Текст исходной программы

|  |  |  |  |
| --- | --- | --- | --- |
| **Адрес** | **Код команды** | **Мнемоника** | **Комментарий** |
| 0D1 | 0200 | CLA | Очистка AC |
| 0D2 | EE19 | ST R | Очистка результата R |
| 0D3 | AE16 | LD Y | Загрузка Y |
| 0D4 | 0700 | INC | и Y+1 |
| 0D5 | 0C00 | PUSH | Кладём в стек Y+1 |
| 0D6 | D661 | CALL $F | Вызываем функцию F(Y+1) |
| 0D7 | 0800 | POP | Выгрузка F(Y+1) функции в AC |
| 0D8 | 4E13 | ADD R | Сложение AC и R: AC = F(Y+1) |
| 0D9 | EE12 | ST R | Сохранение F(Y+1) в R |
| 0DA | AE0E | LD Z | Загрузка Z в AC |
| 0DB | 0740 | DEC | и Z – 1 |
| 0DC | 0C00 | PUSH | Кладём в стек Z – 1 |
| 0DD | D661 | CALL $F | Вызываем функцию F(Z – 1) |
| 0DE | 0800 | POP | Выгрузка F(Z – 1) функции в AC |
| 0DF | 4E0C | ADD R | Сложение AC и R: AC = F(Y+1) + F(Z – 1) |
| 0E0 | EE0B | ST R | Сохранение F(Y+1) + F(Z – 1) в R |
| 0E1 | AE09 | LD X | Загрузка X |
| 0E2 | 0C00 | PUSH | Кладём в стек X |
| 0E3 | D661 | CALL $F | Вызываем функцию F(X) |
| 0E4 | 0800 | POP | Выгрузка F(X) функции в AC |
| 0E5 | 0700 | INC | F(X)+1 |
| 0E6 | 4E05 | ADD R | Сложение AC и R: AC = F(Y+1) + F(Z – 1) + F(X)+1 |
| 0E7 | EE04 | ST R | Сохранение F(Y+1) + F(Z – 1) + F(X)+1в R |
| 0E8 | 0100 | HLT | ОСТАНОВКА ПРОГРАММЫ |
| 0E9 | ZZZZ | Z: word 0xZZZZ | Значение Z |
| 0EA | YYYY | Y: word 0xYYYY | Значение Y |
| 0EB | XXXX | X: word 0xXXXX | Значение X |
| 0EC | 0A65 | R: word 0x0A65 | Результат R |

Подпрограмма

|  |  |  |  |
| --- | --- | --- | --- |
| **Адрес** | **Код команды** | **Мнемоника** | **Комментарий** |
| 661 | AC01 | F: LD (SP+1) | Загрузка аргумента K |
| 662 | F207 | BMI S | Переход на загрузку A если K отрицательное (знаковое сравнение) |
| 663 | 7E09 | CMP A | Выставление флагов по результату операции K - A |
| 664 | F905 | BGE S | Переход на загрузку A если A >= K  (знаковое сравнение) |
| 665 | 0500 | ASL | Умножение K на 2: A = K\*2 |
| 666 | 0500 | ASL | Умножение AC\*2 на 2: AC = K\*4 |
| 667 | 4C01 | ADD &1 | Прибавление K к AC: AC = K\*5 |
| 668 | 6E05 | SUB B | Вычитание из AC B: AC = K\*5-B |
| 669 | CE01 | JUMP (IP+1) | Переход на сохранение результата |
| 66A | AE02 | S: LD A | Загрузка A |
| 66B | EC01 | ST (ST+1) | Сохранение результата |
| 66C | 0A00 | RET | Возврат |
| 66D | 0B67 | A: word 0x0B67 | Константа A |
| 66E | 0084 | B: word 0x0084 | Константа B |

**Псевдо-программа:**

//создание стека  
struct Node {  
 int data;  
 struct Node\* next;  
};  
  
  
// Функция push добавляет новый элемент со значением data на вершину стека.  
void push(struct Node\*\* head, int data) {  
 // Выделяем память для нового узла  
 struct Node\* newNode = (struct Node\*) malloc(sizeof(struct Node));  
 // Записываем значение в новый узел  
 newNode->data = data;  
 // Связываем новый узел с предыдущим узлом  
 newNode->next = \*head;  
 // Новый узел становится вершиной стека  
 \*head = newNode;  
}  
  
//подпрограмма  
int under\_program() {  
 int arg = head->arg; //вызов аргумента со стека  
  
 int arr[2] = {0x0B67, 0x0084}; // создание массива с двумя элементами типа int  
 int \*const\_num = arr; // создание указателя на начало массива  
  
 if (arg < 0 || arg >= \*const\_num) {  
 push(head, \*const\_num) //кладём элемент на вершину стека  
 return 0;  
 }  
  
 int result = (arg << 2) + arg - \*(const\_num+1); //подсчёт результата  
 push(head, result) //кладём элемент на вершину стека  
 return 0;  
}  
  
//начало основной программы  
struct Node\* head = NULL; //создание стека пустого  
  
int main\_arr[3] = {0xZZZZ, 0xYYYY, 0xXXXX}; // создание массива с двумя элементами типа int  
int \*main\_arg = main\_arr; // создание указателя на начало массива  
int ans = 0; //создание R  
  
  
void (\*program) (void); // определяем указатель на функцию  
program = under\_program; // указатель указывает на функцию under\_program  
  
push(head, (\*main\_arg+0x0001)); //кладём на вершину стека первый элемент массива с учётом +1  
program(); //вызов подпрограммы  
int f\_y\_plus\_one = head->f\_y\_plus\_one; //вызов результата подпрограммы со стека  
ans+= f\_y\_plus\_one; //прибавление результата  
  
push(head, (\*(main\_arg+1)-0x0001)); //кладём на вершину стека второй элемент массива с учётом -1  
program(); //вызов подпрограммы  
int f\_y\_plus\_two = head->f\_y\_plus\_two; //вызов результата подпрограммы со стека  
ans+= f\_y\_plus\_two; //прибавление результата  
  
push(head, (\*(main\_arg+2))); //кладём на вершину стека третий элемент массива  
program(); //вызов подпрограммы  
int f\_y\_plus\_three = head->f\_y\_plus\_three; //вызов результата подпрограммы со стека  
ans+= f\_y\_plus\_three+0x0001; //прибавление результата  
  
printf(ans);

**Описание подпрограммы**

Изображение выглядит как текст

Автоматически созданное описание

**Графическое представление функции**

Изображение выглядит как диаграмма

Автоматически созданное описание

**Описание программы**

Нахождения значения данного выражения

R=F(Y+1) + F(Z – 1) + F(X)+1

**Расположение в памяти БЭВМ программы, исходных данных и результатов:**

0E9-0EB – переменные

0EC – результат

66D – константа A

66E – константа B

**Область представления:**

A, B, X, Y, Z, R – целые знаковые шестнадцатеричные числа

**Область допустимых значений**

A = 0B6716 = 291910

B = 008416 = 13210

F(K) = K\*5-B

F(Y+1) + F(Z – 1) + F(X)+1

Изучим функцию f(k), пользуясь описание подпрограммы. Если мы будет в качестве аргумента подавать в неё числа меньшие нуля или большие или равные 2918 (0x0B67), то функция будет возвращать значение 2918 (0x0B67), а значит на этом промежутке переполнения возникнуть не может. Далее рассмотрим промежуток [0; 2918) или же [0x0000; 0x0B67), проанализировав график, получим деление этого промежутка на два других, так как функция не является непрерывной и мы используем аргументы, принадлежащие кольцу целых чисел. Переполнения на промежутках [0; 27) [27; 2918) или же [0x0000; 0x001B) [0x001B; 0x0B67) возникнуть не может. На отрезке [0; 27) или же [0x0000; 0x001B) мы будем получать флаг N.

Так как основная программа вычисляет следующее выражение:

R = F(Y+1) + F(Z – 1) + F(X)+1

то минимально мы можем получить:

+ () + () + 1 = -395 (> -2^15)

а максимально:

+ + + 1= 43375 (> 2^15 – 1)

В первом случае переполнение невозможно.

Во втором случае переполнение возможно.

В функцию аргументы мы передаем значения Y+1, Z – 1, X.

Значит, ОДЗ:

Изображение выглядит как диаграмма

Автоматически созданное описание

**Тестирования работы подпрограммы**

|  |  |  |  |
| --- | --- | --- | --- |
| **Адрес** | **Код команды** | **Мнемоника** | **Комментарий** |
| 65F | 0000 | R: word 0x0000 | RUSULT |
| 660 | 0000 | X: word 0x0000 | Тестовое значение для функции |
| 661 | A660 | LD X | Загрузка аргумента X |
| 662 | F207 | BMI 07 | Переход на загрузку A если X отрицательное (знаковое сравнение) |
| 663 | 7E09 | CMP A | Выставление флагов по результату операции X - A |
| 664 | F905 | BGE 05 | Переход на загрузку A если X >= A  (знаковое сравнение) |
| 665 | 0500 | ASL | Умножение X на 2: A = X\*2 |
| 666 | 0500 | ASL | Умножение AC\*2 на 2: AC = X\*4 |
| 667 | 4660 | ADD X | Прибавление K к AC: AC = X\*5 |
| 668 | 6E05 | SUB B | Вычитание из AC B: AC = X\*5-B |
| 669 | CE01 | JUMP (IP+1) | Переход на сохранение результата |
| 66A | AE02 | LD A | Загрузка A |
| 66B | E65F | ST R | Сохранение результата |
| 66C | 0100 | HLT | Остановка программы |
| 66D | 0B67 | A: word 0x0B67 | Константа A |
| 66E | 0084 | B: word 0x0084 | Константа B |

**Трассировки:**

-155 26 27

809B, 1A, 1B

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Адр | Знчн | IP | CR | AR | DR | SP | BR | AC | PS | NZVC | Адр | Знчн |
| 0D1 | 0200 | 0D1 | 0000 | 000 | 0000 | 000 | 0000 | 0000 | 004 | 0100 |  |  |
| 0D1 | 0200 | 0D2 | 0200 | 0D1 | 0200 | 000 | 00D1 | 0000 | 004 | 0100 |  |  |
| 0D2 | EE19 | 0D3 | EE19 | 0EC | 0000 | 000 | 0019 | 0000 | 004 | 0100 | 0EC | 0000 |
| 0D3 | AE16 | 0D4 | AE16 | 0EA | 001A | 000 | 0016 | 001A | 000 | 0000 |  |  |
| 0D4 | 0700 | 0D5 | 0700 | 0D4 | 0700 | 000 | 00D4 | 001B | 000 | 0000 |  |  |
| 0D5 | 0C00 | 0D6 | 0C00 | 7FF | 001B | 7FF | 00D5 | 001B | 000 | 0000 | 7FF | 001B |
| 0D6 | D661 | 661 | D661 | 7FE | 00D7 | 7FE | D661 | 001B | 000 | 0000 | 7FE | 00D7 |
| 661 | AC01 | 662 | AC01 | 7FF | 001B | 7FE | 0001 | 001B | 000 | 0000 |  |  |
| 662 | F207 | 663 | F207 | 662 | F207 | 7FE | 0662 | 001B | 000 | 0000 |  |  |
| 663 | 7E09 | 664 | 7E09 | 66D | 0B67 | 7FE | 0009 | 001B | 008 | 1000 |  |  |
| 664 | F905 | 665 | F905 | 664 | F905 | 7FE | 0664 | 001B | 008 | 1000 |  |  |
| 665 | 0500 | 666 | 0500 | 665 | 001B | 7FE | 0665 | 0036 | 000 | 0000 |  |  |
| 666 | 0500 | 667 | 0500 | 666 | 0036 | 7FE | 0666 | 006C | 000 | 0000 |  |  |
| 667 | 4C01 | 668 | 4C01 | 7FF | 001B | 7FE | 0001 | 0087 | 000 | 0000 |  |  |
| 668 | 6E05 | 669 | 6E05 | 66E | 0084 | 7FE | 0005 | 0003 | 001 | 0001 |  |  |
| 669 | CE01 | 66B | CE01 | 669 | 066B | 7FE | 0001 | 0003 | 001 | 0001 |  |  |
| 66B | EC01 | 66C | EC01 | 7FF | 0003 | 7FE | 0001 | 0003 | 001 | 0001 | 7FF | 0003 |
| 66C | 0A00 | 0D7 | 0A00 | 7FE | 00D7 | 7FF | 066C | 0003 | 001 | 0001 |  |  |
| 0D7 | 0800 | 0D8 | 0800 | 7FF | 0003 | 000 | 00D7 | 0003 | 001 | 0001 |  |  |
| 0D8 | 4E13 | 0D9 | 4E13 | 0EC | 0000 | 000 | 0013 | 0003 | 000 | 0000 |  |  |
| 0D9 | EE12 | 0DA | EE12 | 0EC | 0003 | 000 | 0012 | 0003 | 000 | 0000 | 0EC | 0003 |
| 0DA | AE0E | 0DB | AE0E | 0E9 | 809B | 000 | 000E | 809B | 008 | 1000 |  |  |
| 0DB | 0740 | 0DC | 0740 | 0DB | 0740 | 000 | 00DB | 809A | 009 | 1001 |  |  |
| 0DC | 0C00 | 0DD | 0C00 | 7FF | 809A | 7FF | 00DC | 809A | 009 | 1001 | 7FF | 809A |
| 0DD | D661 | 661 | D661 | 7FE | 00DE | 7FE | D661 | 809A | 009 | 1001 | 7FE | 00DE |
| 661 | AC01 | 662 | AC01 | 7FF | 809A | 7FE | 0001 | 809A | 009 | 1001 |  |  |
| 662 | F207 | 66A | F207 | 662 | F207 | 7FE | 0007 | 809A | 009 | 1001 |  |  |
| 66A | AE02 | 66B | AE02 | 66D | 0B67 | 7FE | 0002 | 0B67 | 001 | 0001 |  |  |
| 66B | EC01 | 66C | EC01 | 7FF | 0B67 | 7FE | 0001 | 0B67 | 001 | 0001 | 7FF | 0B67 |
| 66C | 0A00 | 0DE | 0A00 | 7FE | 00DE | 7FF | 066C | 0B67 | 001 | 0001 |  |  |
| 0DE | 0800 | 0DF | 0800 | 7FF | 0B67 | 000 | 00DE | 0B67 | 001 | 0001 |  |  |
| 0DF | 4E0C | 0E0 | 4E0C | 0EC | 0003 | 000 | 000C | 0B6A | 000 | 0000 |  |  |
| 0E0 | EE0B | 0E1 | EE0B | 0EC | 0B6A | 000 | 000B | 0B6A | 000 | 0000 | 0EC | 0B6A |
| 0E1 | AE09 | 0E2 | AE09 | 0EB | 001B | 000 | 0009 | 001B | 000 | 0000 |  |  |
| 0E2 | 0C00 | 0E3 | 0C00 | 7FF | 001B | 7FF | 00E2 | 001B | 000 | 0000 | 7FF | 001B |
| 0E3 | D661 | 661 | D661 | 7FE | 00E4 | 7FE | D661 | 001B | 000 | 0000 | 7FE | 00E4 |
| 661 | AC01 | 662 | AC01 | 7FF | 001B | 7FE | 0001 | 001B | 000 | 0000 |  |  |
| 662 | F207 | 663 | F207 | 662 | F207 | 7FE | 0662 | 001B | 000 | 0000 |  |  |
| 663 | 7E09 | 664 | 7E09 | 66D | 0B67 | 7FE | 0009 | 001B | 008 | 1000 |  |  |
| 664 | F905 | 665 | F905 | 664 | F905 | 7FE | 0664 | 001B | 008 | 1000 |  |  |
| 665 | 0500 | 666 | 0500 | 665 | 001B | 7FE | 0665 | 0036 | 000 | 0000 |  |  |
| 666 | 0500 | 667 | 0500 | 666 | 0036 | 7FE | 0666 | 006C | 000 | 0000 |  |  |
| 667 | 4C01 | 668 | 4C01 | 7FF | 001B | 7FE | 0001 | 0087 | 000 | 0000 |  |  |
| 668 | 6E05 | 669 | 6E05 | 66E | 0084 | 7FE | 0005 | 0003 | 001 | 0001 |  |  |
| 669 | CE01 | 66B | CE01 | 669 | 066B | 7FE | 0001 | 0003 | 001 | 0001 |  |  |
| 66B | EC01 | 66C | EC01 | 7FF | 0003 | 7FE | 0001 | 0003 | 001 | 0001 | 7FF | 0003 |
| 66C | 0A00 | 0E4 | 0A00 | 7FE | 00E4 | 7FF | 066C | 0003 | 001 | 0001 |  |  |
| 0E4 | 0800 | 0E5 | 0800 | 7FF | 0003 | 000 | 00E4 | 0003 | 001 | 0001 |  |  |
| 0E5 | 0700 | 0E6 | 0700 | 0E5 | 0700 | 000 | 00E5 | 0004 | 000 | 0000 |  |  |
| 0E6 | 4E05 | 0E7 | 4E05 | 0EC | 0B6A | 000 | 0005 | 0B6E | 000 | 0000 |  |  |
| 0E7 | EE04 | 0E8 | EE04 | 0EC | 0B6E | 000 | 0004 | 0B6E | 000 | 0000 | 0EC | 0B6E |
| 0E8 | 0100 | 0E9 | 0100 | 0E8 | 0100 | 000 | 00E8 | 0B6E | 000 | 0000 |  |  |
| 0E9 | 809B | 09E | 0100 | 0E8 | 0100 | 000 | 00E8 | 0B6E | 000 | 0000 |  |  |
| 09E | 809B | 09F | 0100 | 09E | 809B | 000 | 00E8 | 0B6E | 000 | 0000 | 09E | 809B |
| 09F | 0000 | 0E9 | 0100 | 09E | 809B | 000 | 00E8 | 0B6E | 000 | 0000 |  |  |
| 0E9 | 809B | 0EA | 0100 | 0E9 | 809B | 000 | 00E8 | 0B6E | 000 | 0000 | 0E9 | 809B |

Доп. Задание:

|  |  |  |  |
| --- | --- | --- | --- |
| **Адрес** | **Код команды** | **Мнемоника** | **Комментарий** |
| 0x512 | 0524 | A: word 0x0524 | Адрес начала массива значений функции |
| 0x513 | 0000 | P: word 0x0000 | аргумент |
| 0x514 | 0032 | N: word 0x0032 | Размер |
| 0x515 | 0200 | R: word 0x0200 |  |
| 0x516 | 0200 | CLA | Очистка аккумулятора |
| 0x517 | A513 | LD 513 |  |
| 0x518 | 0C00 | PUSH |  |
| 0x519 | D661 | CALL $F |  |
| 0x51A | 0800 | POP |  |
| 0x51B | EAF6 | ST | 0000 1010 |
| 0x51C | A513 | LD 513 | Текущий арг |
| 0x51D | 0700 | INC | Увеличение аргумента |
| 0x51E | E513 | ST 513 | Запись нового аргумента |
| 0x51F | 8514 | LOOP N | Проверка границы массива  (N – 1 -> N; IF N <= 0: IP+1 -> IP) |
| 0x520 | C516 | JUMP 516 | Прыжок в начало цикла |
| 0x521 | 0100 | HLT | Остановка и переход в пультовый режим |
| 0x522 |  |  |  |
| 0x523 |  |  |  |
| 0x524 |  |  | Первый элемент массива |
| 0x525 |  |  | Второй элемент массива |
| 0x526 |  |  | Третий элемент массива |

**Выводы:**

В процессе выполнения лабораторной работы был получен опыт работы с подпрограммами и стеком.

**Список литературы:**

1. В.В. Кириллов АРХИТЕКТУРА БАЗОВОЙ ЭВМ Учебное пособие / В.В. Кириллов — 1. — Санкт-Петербург: САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ, 2010 — 142 c.