МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования
«Воронежский государственный технический университет»

Методические рекомендации по практическим занятиям

междисциплинарного курса: МДК.02.03 Регистрация основных событий в автоматизированных системах

Специальность: 10.02.05 Обеспечение информационной безопасности автоматизированных систем

Квалификация выпускника: Техник по защите информации

Нормативный срок обучения: 3 года 10 месяцев

Форма обучения: Очная

2021

Методические указания по практическим занятиям междисциплинарного курса: МДК.02.03 Регистрация основных событий в автоматизированных системах разработаны на основе федерального государственного образовательного стандарта по специальности среднего профессионального образования 10.02.15 Обеспечение информационной безопасности автоматизированных систем Утвержденным приказом Минобрнауки России от 09.12.2016г. №1553

дата утверждения и №)

Методические указания рассмотрены на заседании методического совета СПК и рекомендованы к использованию

«19» 02. 2020 года Протокол № __1__

Председатель методического совета СПК

Сергеева Светлана Ивановна _____

Методические указания утверждены на заседании педагогического совета СПК «28» 02. 2020 года Протокол № _6__

Председатель педагогического совета СПК

Облиенко Алексей Владимирович _____

Организация-разработчик: ФГБОУ ВО «ВГТУ»

Разработчики:

__ Демихова Ирина Владимировна _____

(Ф.И.О., ученая степень, звание, должность

_____

(Ф.И.О., ученая степень, звание, должность)

_____

(Ф.И.О., ученая степень, звание, должность)

_____

(Ф.И.О., ученая степень, звание, должность)

ПРАКТИЧЕСКАЯ РАБОТА № 1

Изучение работы программы-эмулятора EMU8086.

Цель: Изучение принципов построения и работы программной модели эмулятора универсально однокристального микропроцессора.

Теоретические вопросы

1. Опишите структуру ассемблерной программы.
2. Опишите процесс обработки программы на языке ассемблера. Каковы этапы получения выполняемого файла?
3. Опишите назначение эмулятора Emu8086.
4. Опишите назначение основных компонентов Emu8086..

Задание.

1. Запустить эмулятор Emu8086

2. Изучить программные компоненты Emu8086 (основные окна, кнопки)

3. Выполнить пример, описанный пошагово в теоретическом разделе

4. Самостоятельно по образцу, рассмотренному выше, создать первую программу, которая будет выводить на экран: ‘Hello World’

5. Записать код выполненной программы, дать комментарии каждому выполняемому блоку команд

6. Изучить режимы работы отладчика

ПРАКТИЧЕСКАЯ РАБОТА № 2

Операции со знаковыми и беззнаковыми величинами

1 Учет знаков в операциях

Для записи отрицательного числа в программе на ассемблере используется символ ‘-’, например:

x **db** -5

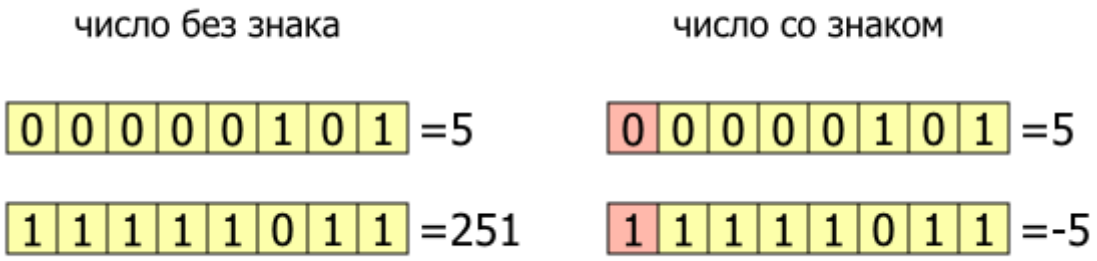
y **db** -25h

z **db** -77o

k **db** -101b

Со знаковыми и беззнаковыми числами нужно быть внимательным, потому что только вы знаете, какие числа используются в вашей программе! Процессору абсолютно все равно, какие данные он обрабатывает, поэтому невнимательность может привести к ошибке. Один и тот же байт может

интерпретироваться по-разному, в зависимости от того со знаком число или без. Например, числу со знаком -5 соответствует число без знака 251:



Диапазоны значений чисел со знаком и без знака

При программировании на ассемблере (как, впрочем, и на многих других языках) необходимо учитывать ещё один важный момент. А именно – ограничение диапазона представления чисел. Например, если размер беззнаковой переменной равен 1 байт, то она может принимать всего 256 различных значений. Это означает, что мы не сможем представить с её помощью число, больше 255 (11111111_2). Для такой же переменной со знаком максимальным значением будет 127 (01111111_2), а минимальным -128 (10000000_2). Аналогично определяется диапазон для 2- и 4-байтных переменных.

2. Вычитание беззнаковых величин

Вычитание выполняется с помощью команды SUB (subtract — вычесть). В остальном все этапы выполнения вычисления повторяют действия, которые были описаны для операции сложения.

В регистр AX заносится уменьшаемое, а в регистр BX — вычитаемое. Результат выполнения инструкции появится в регистре AX.

Команда SUB имеет следующий формат: SUB приемник, источник

Примеры:

mov al,10

sub al,7 ---> al = 3; al - приемник, 7 - источник

mov ax,25000

sub ax,10000 ---> ax = 15000; ax - приемник, 10000 - источник

mov cx,100

mov bx,15

sub cx,bx ---> cx = 85, bx = 15 (bx не меняется); cx - приемник, bx – источник

Задание 1.

Проверьте, как микропроцессор использует форму двоичного дополнения для представления отрицательных результатов. Выполните вычитание из нуля единицы (т.е., 0 – 1). Какой результат получен и почему?

Команда INC

Команда INC увеличивает на единицу регистр. Она эквивалентна команде

ADD источник, 1

только выполняется быстрее.

Примеры:

mov al,15

inc al ---> теперь AL = 16 (эквивалентна add al,1)

mov dh,39h

inc dh ---> DH = 3Ah (эквивалентна add dh,1)

mov cl,4Fh

inc cl ---> CL = 50h (эквивалентна add cl,1)

Команда DEC

Оператор dec уменьшает значение приемника на 1

Пример:

mov ah,12 ---> AH=12

dec ah ---> AH=11

3. Операции с байтами

В микропроцессорах Intel используются двухбайтовые машинные слова. Каждый регистр общего назначения (AX, BX, CX и DX) может хранить одно машинное слово. Однако имеется возможность оперировать с отдельными байтами этих регистров. В этом случае каждый регистр рассматривается состоящим из старшего (High) и младшего (Low) байтов. Обозначения отдельных байтов из регистров состоят из двух букв. Первая задает имя регистра (A, B, C или D), а вторая указывает, какой это байт регистра. Для обозначения старшего байта используется буква H, а младшего — L. Таким образом, регистр AX можно рассматривать, состоящим из двух однобайтовых регистров AH и AL.

Микропроцессор может выполнять арифметические операции над отдельными байтами.

Задание 2.

Введите в регистр AX число 0102h (два байта) и выполните инструкцию

ADD AH, AL

Сделать выводы каков результат выполнения операции, который будет помещен в регистр AH?

4. Умножение беззнаковых величин

Умножение двух 16-битных чисел может дать 32-разрядный результат, поэтому инструкция умножения MUL (multiply — умножить) размещает результат в двух регистрах DX и AX. Старшие 16 бит помещаются в регистр DX, а младшие в AX.

Замечание.

При выполнении операции умножения одним из множителей всегда является значение из регистра AX.

Задание 3.

Выполните умножение чисел 7C4Bh (в регистр AX) и 100h (BX).

Сделать выводы каков результат операции и почему?

5. Деление беззнаковых величин

Команды микропроцессора предназначены для выполнения целочисленных операций. Так как деление целых чисел нацело происходит далеко не всегда, то результат деления формируется из двух целых чисел — частного и остатка от деления.

Делимое всегда помещается в пару регистров AX, DX, поэтому в инструкции деления DIV (divide — делить) необходимо указать только регистр с делителем. После выполнения деления регистр AX будет содержать частное, а регистр DX — остаток.

Задание 4.

Выполните деление числа 7C4B12h (DX=007Ch, AX=4B12h) на 0100h (BX).

Каков результат выполнения операции и почему?

Пример программы

Чтобы всё стало совсем понятно, напишем небольшую программу. Требуется вычислить значение формулы: *e=a-(b+c-1)+(-d)*. Все числа являются 8-битными целыми со знаком. Объявим их после кода и придумаем какие-нибудь значения.

```
1  use16 ;Генерировать 16-битный код
2  org 100h ;Программа начинается с адреса 100h
3
4  mov al,[a] ;Загружаем значение a в AL
5  mov ah,[b] ;Загружаем значение b в AH
6  add ah,[c] ;AH = AH + c = b+c
7  dec ah ;AH = AH - 1 = b+c-1
```

```
8  sub al,ah ;AL = AL - AH = a-(b+c-1)
9  mov cl,[d] ;CL = d
10 neg cl ;CL = -CL = -d
11 add al,cl ;AL = AL + CL = a-(b+c-1)+(-d)
12 mov [e],al ;Сохраняем результат в e
13
14 mov ax,4C00h ;\
15 int 21h ;/ Завершение программы
16 ;-------------------------------------------------------
17 a db 2
18 b db 3
19 c db 5
20 d db -8
21 e db ?
```

Квадратные скобки означают, что операнд находится по адресу, указанному внутри этих скобок. Так как вместо имени переменной эмулятор подставляет её адрес, то такая запись позволяет прочитать или записать значение переменной.

Задание 5

Напишите программу для вычисления формулы *k=m+1-(n-1-r)*. Все числа 16-битные целые со знаком. Запустите в отладчике и проверьте правильность вычисления.

Задание 6

Разработайте программу, реализующую указанную формулу, исполнить программу с несколькими наборами исходных данных, проверить правильность результатов.

+

1. X= A - 5 (B - 2C) + 2
2. X= - 4A + (B + C) / 4 + 2
3. X= 7A - 2B - 100 + C
4. X= - A / 2 + 4 (B + 1) + 3C
5. X= 5 (A - B) - 2C + 5
6. X= (A/ 2 + B) / 4 + C - 1
7. X= - (C + 2A + 4B + B)
8. X= 6C + (B - C + 1) / 2
9. X= 2 - B (A + B) + C / 4
10. X= 2B - 1 + 4 (A - 3C)
11. X= (2A + B) / 4 - C / 2 + 168
12. X= 6 (A - 2B + C / 4) + 10
13. X= 5 (A - B) + C * 4
14. X= - (- (C + 2A) * 4B + 38)
15. X= A - 3 (A + B) + C * 4
16. X= 3(A - 2B) +50 – C / 2
17. X= (3A + 2B) - C / 4 + 217
18. X= 3(C - 2A) + (B - C + 1) / 2
19. X= (2A + B) / 4 - C / 2 + 168
20. X= 6 (A - 2B + C / 4) + 10
21. X= 3 (A - 4B) + C / 4

22. X= - (- (C + 2A) * 5B - 27)
23. X= A / 2 - 3 (A + B) + C * 4
24. X= 3(A - 2B) +50 – C / 2
25. X= 5A + 2B - B / 4 + 131

ПРАКТИЧЕСКАЯ РАБОТА № 3

Изучение процесса создания программ на языке Ассемблера.

Цель работы: Изучение процесса создания програм

Теоретическое обоснование

Учебный микропроцессорный комплект (УМК) создан на базе микропроцессора КР580ВМ80А и предназначен для подготовки специалистов в области микропроцессорной техники и обучению основам программирования.

Микропроцессор КР580ВМ80А предназначен для выполнения определенного набора команд (представленного в приложении 1) и реализован на одной БИС.

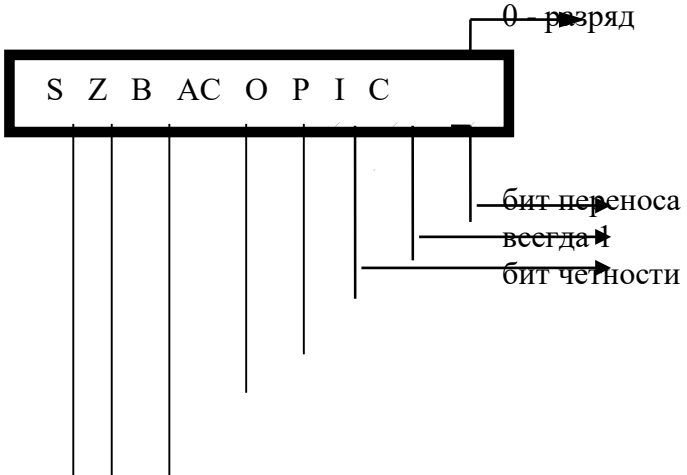
Все команды микропроцессора можно разделить на 8 основных групп:

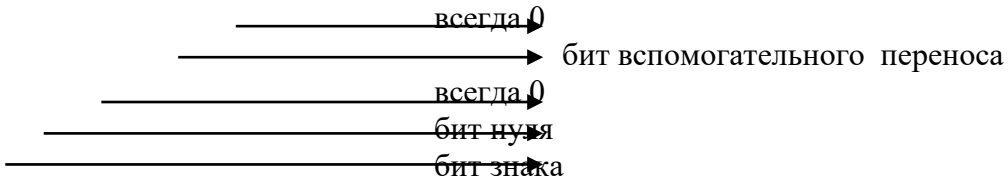
- команды пересылки данных и загрузки регистров;
- команды работы с памятью;
- арифметические, логические команды и команды сравнения и сдвига;
- команды передачи управления;
- команды манипуляции стеком;
- команды вызова подпрограмм и возврата из них;
- команды ввода/вывода;
- команды обработки прерываний.

Для оперативного хранения данных в микропроцессоре имеется семь 8-разрядных регистров общего назначения:

A - регистр-накопитель, аккумулятор;
B - регистр общего назначения;
C - регистр общего назначения;
D - регистр общего назначения;
E - регистр общего назначения;
H - регистр косвенного адреса;
L - регистр косвенного адреса.

Для хранения битов условий имеется 8-разрядный регистр признаков PSW. Распределение и назначение битов условий регистров PSW следующее:





Бит переноса устанавливается в 1, если возникает переполнение старшего разряда в байте, в противном случае устанавливается в 0.

Бит четности устанавливается в 1, если количество единиц в байте четное, в противном случае устанавливается в 0.

Бит вспомогательного переноса устанавливается в 1, если возникает перенос из младшей тетрады байта в старшую, в противном случае - в 0.

Бит нуля устанавливается в 1, если результат выполнения операции равен 0, в противном случае в 0.

Бит знака устанавливается в 1, если результат операции отрицательный (ст. разряд байта равен 1), в противном случае, устанавливается в 0.

16-разрядный регистр-указатель стека SP служит для определения нижней границы области стека.

16-разрядный регистр-счетчик команд PC служит для хранения адреса выполненной команды.

Содержание работы:

1.1. Ознакомление с органами управления, ввода, индикации УМК.
1.2. Включение и запуск УМК.
1.3. Команды системного монитора УМК.
1.3.1. Просмотр и модификация содержимого ячейки памяти.
1.3.2. Просмотр и модификация содержимого регистра микропроцессора (МП).
1.3.3. Старт программы.
1.3.4. Заполнение массива памяти константой.
1.3.5. Подсчет контрольной суммы.
1.3.6. Перемещение массива памяти.

Задание:

1. Заполните (и проверьте) массив памяти с адресом 840H - 84FH данными - FFH.
2. Заполните (и проверьте) массив памяти с адресом 93AH - 952H данными 05H.
3. Заполните (и проверьте) массив памяти с адресом 960H - 97FH данными C7H.

ПРАКТИЧЕСКАЯ РАБОТА № 4

Операции ввода/вывода в Ассемблере.

Цель: освоить принципы взаимодействия микропроцессора с внешними устройствами, основанные на командах ввода и вывода данных. В качестве примера изучить работу программы-драйвера, осуществляющей приём данных с клавиатуры и вывод их на дисплей.

Теоретические вопросы:

1. Для чего служит регистр-аккумулятор процессора?
2. С какими внешними устройствами может взаимодействовать микропроцессор?
3. Какие порты и с какой целью используются в программе?

4. Для чего предназначен буфер регенерации?

Задание 1. Запишите в память коды программы ввода данных с клавиатуры и отображения их на экране дисплея

Задание 2. Осуществить считывание данных из внешнего порта, адрес которого 0x80

Задание 3. Осуществить вывод данных из регистра-аккумулятора микропроцессора во внешний порт, адрес которого 0x89

Задание 4. Создать программу-драйвер, обеспечивающую взаимодействие микропроцессора с периферийной техникой.

Задание 5. Составить программу-драйвер в соответствии с таблицей:

PORT A	порт номера индикатора содержит: 20H - 0 индикатор (самый правый) 10H- 1 индикатор 08H- 2 индикатор 04H- 3 индикатор 02H- 4 индикатор 01H- 5 индикатор Адрес: PORTA = F8H
PORT B	порт вывода данных - выводится код отображаемого символа. Адрес: PORTB = F9H
PORT C	порт состояния клавиатуры. Адрес: PORTC = FAH

ПРАКТИЧЕСКАЯ РАБОТА № 5

Программирование линейных алгоритмов на языке ASSEMBLER

Цель: реализовать программу для вычисления простой формулы на языке Ассемблер.

Теоретические вопросы

1. Виды алгоритмов.
2. Элементы графического алгоритма.
3. Простейшие команды.
4. Синтаксис программы.

Задание. Разработать программу, реализующую указанную формулу.

Запустить программу с тремя – четырьмя наборами исходных данных и проверить правильность результатов.

Варианты заданий:

1 X=A-5(B-2C)+2

14 X=-(-(C+2A)4B+38)

2 X=-4A+(B+C)/4+2
15 X=A-3(A+B)+C mod 4
3 X=7A-2B-100+C
16 X=3(A-2B)+50-C/2
4 X=-A/2+4(B+1)+3C
17 X=(3A+2B)-C/4+217
5 X=5(A-B)-2C+5
18 X=3(C-2A)+(B-C+1)/2
6 X=(A/2+B)/4+C-1
19 X=(2A+B)/4-C/2+168
7 X=-(C+2A+4B+B)
20 X=6(A-2B+C/4)+10
8 X=6C+(B-C+1)/2
21 X=3(A-4B)+C/4
9 X=2-B(A+B)+C/4
22 X=-(-(C+2A)5B-27)
10 X=2B-1+4(A-3C)
23 X=A/2-3(A+B)+C*4
11 X=(2A+B)/4-C/2+168
24 X=3(A-2B)+50-C/2
12 X=6(A-2B+C/4)+10
25 X=5A+2B-B/4+131
13 X=5(A-B)+Cmod 4

ПРАКТИЧЕСКАЯ РАБОТА № 6

Программирование разветвляющихся алгоритмов на языке ASSEMBLER

Цель работы: Освоить практические навыки программирования разветвляющихся алгоритмов на языке Ассемблер

Задания для самостоятельного выполнения:

1. Составить программу для индивидуального задания на основе приведенного примера
2. Выполнить программу в среде EMU8086.
3. Выполнить программу в двух возможных случаях при X≥N и X<N.
4. Составить аналогичную программу на языке C++ и проверить полученные результаты.

Порядок выполнения работы

Рассмотреть пример приведенный ниже, изучить структуру программы, используемые команды и функции и использовать их при решении индивидуального задания.

Пример. Составить программу на языке Ассемблер для вычисления кусочно-заданной

функции:

```
;GOTOXY 10, 2                ; установить курсор на 2 ряд и
                               10 столбец

LEA     SI, msgX             ; Запрос на ввод          X
CALL    print_string         ;
CALL    scan_num             ; ввод числа в CX.
MOV     X,CX
;GOTOXY 10,3
LEA     SI, msgA             ; Запрос на ввод          A
CALL    print_string         ;
CALL    scan_num             ; ввод числа в CX. MOV
        A,CX
;GOTOXY 10,4
LEA     SI, msgB             ; Запрос на ввод B CALL
        print_string         ;
CALL    scan_num             ; ввод числа в CX. MOV
      B,CX
;GOTOXY 10,5
LEA     SI, msgC             ; Запрос на ввод C CALL
        print_string         ;
CALL    scan_num             ; ввод числа в CX. MOV
        C,CX
;GOTOXY 10,6
LEA     SI, msgD             ; Запрос на ввод D CALL
        print_string         ;
CALL    scan_num             ; ввод числа в CX. MOV
      D,CX
CMP X,2
JGE VAG1
MOV AX,C
IMUL    X
ADD AX,D
JMP VAG
VAG1: MOV AX,A
IMUL X
SUB AX,B
VAG: ;GOTOXY 10,8
CALL    pthis
DB    13, 10, 'Otvet: ', 0
```

```
CALL        print_num                   ; Ввод числа в AX.
RET                                     ; Возврат в операционную систему. msgX          DB
            'Vvedite X : ', 0
msgA DB 'Vvedite A : ', 0 msgB DB 'Vvedite B
: ', 0 msgC   DB  'Vvedite C :' , 0
msgD        DB     'Vvedite D : ', 0 X        DW?
A           DW    ?
B           DW    ?
C           DW    ?
D           DW    ?
Y           DW    ?
DEFINE_SCAN_NUM DEFINE_PRINT_STRING
DEFINE_PRINT_NUM
DEFINE_PRINT_NUM_UNS          ; Требуется для print_num.
DEFINE_PTHIS
DEFINE_CLEAR_SCREEN
END                         ; Конец компиляции.
```

1. variant $X \geq 2$

```
Vvedite     X :    3
Vvedite     A :    2
Vvedite     B :    2
Vvedite     C :    3
Vvedite     D :    4
Otvet: 4
```

2. variant $X < 2$

```
Vvedite     X :    1
Vvedite     A :    2
Vvedite     B :    2
Vvedite     C :    3
Vvedite     D :    4
Otvet: 7
```

Индивидуальные задания

№	Выражение	Данные				
		A	B	C	D	X
1.	$Y=\begin{cases}AX^2-B, & \text{если } X\geq 4\\ CX^2+D, & \text{если } X<4\end{cases}$	1	2	5	3	7 и 2
2.	$Y=\begin{cases}AX^2-B, & \text{если } X\geq 6\\ CX+D, & \text{если } X<6\end{cases}$	2	3	4	2	10 и 3
3.	$Y=\begin{cases}AX+B^2, & \text{если } X\geq 6\\ C^2X+D, & \text{если } X<6\end{cases}$	1	1	4	5	8 и 3
4.	$Y=\begin{cases}A^2X-B, & \text{если } X\geq 8\\ C^2X+D, & \text{если } X<8\end{cases}$	2	1	4	5	4 и 10
5.	$Y=\begin{cases}AX+7B, & \text{если } X\geq 10\\ CX-4D, & \text{если } X<10\end{cases}$	1	2	3	5	16 и 4
6.	$Y=\begin{cases}AX-B^2, & \text{если } X\geq 8\\ CX+12D, & \text{если } X<8\end{cases}$	1	2	3	1	9 и 3
7.	$Y=\begin{cases}2AX-B^2, & \text{если } X\geq 2\\ CX+D, & \text{если } X<2\end{cases}$	6	4	3	1	4 и 1
8.	$Y=\begin{cases}AX-8B, & \text{если } X\geq -6\\ 7CX+4D, & \text{если } X<-6\end{cases}$	1	2	1	4	0 и -9
9.	$Y=\begin{cases}AX-7B, & \text{если } X\geq 7\\ CX+12D, & \text{если } X<7\end{cases}$	2	4	5	1	9 и 3
10.	$Y=\begin{cases}AX-B, & \text{если } X\geq 5\\ C^2X+5D, & \text{если } X<5\end{cases}$	8	1	2	1	7 и 2
11.	$Y=\begin{cases}A^2X^2-B, & \text{если } X\geq 3\\ CX+D, & \text{если } X<3\end{cases}$	2	1	6	5	6 и 2
12.	$Y=\begin{cases}A+XB, & \text{если } X\geq 13\\ C+XD, & \text{если } X<13\end{cases}$	1	1	4	5	14 и 10
13.	$Y=\begin{cases}AX-B, & \text{если } X\geq 14\\ CX+D, & \text{если } X<14\end{cases}$	5	1	4	2	18 и 12
14.	$Y=\begin{cases}AX-B^2, & \text{если } -X\geq 20\\ CX+D^2, & \text{если } -X<20\end{cases}$	2	1	4	2	23 и 6
15.	$Y=\begin{cases}AX-B, & \text{если } X\geq 24\\ CX-D^2, & \text{если } X<24\end{cases}$	2	1	3	4	30 и 20
16.	$Y=\begin{cases}AX+B, & \text{если } X\geq 10\\ 5CX-D, & \text{если } X<10\end{cases}$	1	2	4	2	12 и 8
17.	$Y=\begin{cases}AX+2B, & \text{если } X\geq 4\\ CX^2-D, & \text{если } X<4\end{cases}$	3	2	4	1	6 и 3
18.	$Y=\begin{cases}A^2X-3B, & \text{если } X\geq 5\\ CX-D^2, & \text{если } X<5\end{cases}$	5	1	5	2	10 и 4
19.	$Y=\begin{cases}2AX+B, & \text{если } X\geq 4\\ CX+2D, & \text{если } X<4\end{cases}$	2	3	1	1	8 и 2
20.	$Y=\begin{cases}AX-3B, & \text{если } X\geq 6\\ CX+D, & \text{если } X<6\end{cases}$	4	1	3	4	10 и 4
21.	$Y=\begin{cases}3AX-B, & \text{если } X\geq 12\\ 4CX-D, & \text{если } X<12\end{cases}$	3	1	4	1	14 и 8
22.	$Y=\begin{cases}AX-4B, & \text{если } X\geq 18\\ CX-2D, & \text{если } X<18\end{cases}$	5	1	2	1	20 и 16
23.	$Y=\begin{cases}2AX-B^2, & \text{если } X\geq 16\\ CX-3D, & \text{если } X<16\end{cases}$	5	1	3	2	18 и 10
24.	$Y=\begin{cases}A^2X-B, & \text{если } X\geq 8\\ 3CX+D, & \text{если } X<8\end{cases}$	4	1	1	1	10 и 6
25.	$Y=\begin{cases}A^2X-B, & \text{если } X\geq 16\\ CX^2-4D, & \text{если } X<16\end{cases}$	5	1	4	2	18 и 12

ПРАКТИЧЕСКАЯ РАБОТА № 7

Программирование циклических алгоритмов на языке ASSEMBLER.

Цель: Освоить практические навыки программирования и выполнения циклических алгоритмов на языке Ассемблер.

Задания для самостоятельной работы

1. На основе приведенного примера составить программу для индивидуального задания.
2. Выполнить программу в среде EMU8086.
3. Составить аналогичную программу на языке C++ и сравнить результаты.

Порядок выполнения

Рассмотреть пример приведенный ниже, изучить структуру программы, применяемые команды и функции, и использовать их при решении индивидуального задания.

Пример. Составить программу на языке Assembler для вычисления суммы значений

функции $y=f(x)$ на отрезке [a,b] с шагом h и выполнить в среде EMU8086.

$$S = \sum_{x=a}^{x=b} f(x),$$

где x=a,a+h, a+2h.....

Текст программы:

```
include 'emu8086.inc'
```

```
;s= SUM(y=2x+7) where x=(a to b) with step H ORG     100h
GOTOXY 10,3
LEA       SI, msgA                ; запрос числа              ACALL     print_string    ;
CALL      scan_num                ; запись A в CX.
MOV     A,CX
GOTOXY 10,4
LEA       SI, msgB                ; запрос числа              ACALL     print_string    ;
CALL      scan_num                ; запись B в CX. MOV
          B,CX
GOTOXY 10,5
LEA       SI, msgh                ; запрос числа              HCALL     print_string    ;
CALL      scan_num                ; запись H в CX. MOV
          H,CX
MOV bX,A
  MOV
  X,BX
vag:MOV AX,X
IMUL      DD
ADD ax,7
MOV bx,h
ADD x,bx
ADD s,ax
MOV bx,b
CMP x,bx
JLE vag
MOV ax,s
GOTOXY 10,8
CALL      pthis
DB    13, 10, 'Netice: ', 0
CALL      print_num               ; запись числа в AX.
RET                               ; Возврат в операционную систему. msgA      DB
          'Vvedite A : ', 0
msgB      DB    'Vvedite B: ', 0 msgH   DB
          'Vvedite H :' , 0 X     DW    ?
A         DW    2
B         DW    6
H         DW    1
F         DW    ?
S         DW    0
DD        DW                      2
DEFINE_SCAN_NUM DEFINE_PRINT_STRING DEFINE_PRINT_NUM
DEFINE_PRINT_NUM_UNS              ; необходимо для print_num. DEFINE_PTHIS
DEFINE_CLEAR_SCREEN
END       ; Конец компиляции.
```

Задания для индивидуального выполнения:

№	Выражение	Данные		
		A	B	H
1.	Y=3X-8	1	6	1
2.	Y=7X+5	2	5	1
3.	Y=4X-6	1	6	2
4.	Y=5X+2	2	5	1
5.	Y=3X-6	1	5	1
6.	Y=4X+7	0	6	2
7.	Y=2X+3	1	6	2
8.	Y=3X+8	1	4	1
9.	Y=3X-2	2	4	1
10.	Y=5X+1	1	6	1
11.	Y=2X+3	2	8	2
12.	Y=4X+2	1	6	1
13.	Y=2X-5	1	6	1
14.	Y=4X+1	2	8	2
15.	Y=2X+3	2	10	4
16.	Y=2X+3	2	10	4
17.	Y=2X+3	2	10	4
18.	Y=2X+3	2	10	4
19.	Y=2X+3	2	10	4
20.	Y=2X+3	2	10	4
21.	Y=2X+3	2	10	4
22.	Y=2X+3	2	10	4
23.	Y=2X+3	2	10	4
24.	Y=2X+3	2	10	4
25.	Y=2X+3	2	10	4

ПРАКТИЧЕСКАЯ РАБОТА № 8

Программирование ветвлений и циклов.

Цель: Получить практические навыки программирования и выполнения ветвлений и циклических алгоритмов на языке Ассемблер.

Пример выполнения работы

Дан массив из десяти слов, содержащих целые числа. Требуется найти максимальное значение в массиве.

Текст программы:

```
LEABX, MASS
MOVCX, 10
MOV AX, [BX]
BEG: CMP [BX], AX JL
NO
MOV AX, [BX]
NO: INCBX
INCBX
LOOP BEG
MOV MAX, AX
HLT
MAX DW ?
MASS DW 10, 24,76,479, -347, 281, -24, 70, 124, 97
```

Варианты заданий

При сдаче задания, помимо исходного кода программы необходимо представить блок-схему алгоритма. Для составления блок- схемы рекомендуется использовать программу MSVisio.

Дан массив из десяти знаковых чисел (слов или байт). Требуется:

1. Найти количество отрицательных чисел. Массив байт.
2. Найти сумму всех положительных и отрицательных чисел. Массив слов
3. Найти сумму абсолютных величин. Массив байт.
4. Найти количество положительных чисел. Массив байт.
5. Поменять местами пары соседних чисел. Массив слов.
6. Переставить числа в обратном порядке. Массив байт.
7. Заменить все отрицательные числа нулями. Массив байт.
8. Найти среднее арифметическое чисел. Массив слов.
9. Найти количество чисел больших 10h. Массив слов.
10. Найти наименьшее по абсолютной величине числа. Массив байт.
11. Найти наибольшее отрицательное число. Массив байт.
12. Найти произведение положительных элементов последовательности. Массив слов.
13. Найти среднее арифметическое квадратов ненулевых элементов последовательности. Массив слов.
14. Найти полусумму наибольшего и наименьшего чисел. Массив байт.

15. Найти среднее арифметическое отрицательных элементов последовательности. Массив слов.
16. Найти сколько в массиве чисел больше 12h и меньше 0AFh. Массив байт.
17. Найти есть ли в массиве два нуля, идущих подряд. Массив слов.
18. Найти сумму абсолютных величин меньших 6. Массив байт.
19. Найти среднее арифметическое чисел больших 10. Массив слов.
20. Найти сколько чисел равно12h. Массив байт.
21. Заменить все отрицательные числа их модулями. Массив байт.
22. Найти среднее арифметическое положительных чисел Массив слов.
23. Найти количество чисел меньших 10h. Массив байт.
24. Найти наименьшее среди положительных чисел. Массив слов.
Найти наибольшее отрицательное число. Массив байт.

ПРАКТИЧЕСКАЯ РАБОТА № 9

Связь подпрограмм на Ассемблере с программами на языке высокого уровня

Цели: Изучить способы и методы комбинации ассемблера с языками высокого уровня.

Цель. Ход работы

1.Ознакомиться с теоретической частью

2.Ответить на контрольные вопросы

3. Выполнить практическое задание

4.Оформить отчет

Контрольные вопросы

1.Когда может возникать необходимость написания ассемблерных подпрограмм?

2.Как должна выглядеть командная строка для компилятора MASM, если подпрограмма на ассемблере будет применяться в приложении, написанном на Delphi?

3.Что необходимо добавить в проект перед сборкой приложения в Visual C++.NET?

Практическое задание

1. Ознакомиться с теоретическим материалом по данной теме.

2. Изучить и отладить программы из теоретической части.

ПРАКТИЧЕСКАЯ РАБОТА № 10

Способы реверс-инжиниринга кода.

Цель: понять принцип работы или обнаружить недокументированные возможности. Исследовать вредоносные приложения, разбираться как они работают.

Теоретические вопросы

1. Понятие криптографической системы.
2. Классификация криптографических систем.
3. Проблема распределения ключей.
4. Асимметричные алгоритмы шифрования.
5. Типы односторонних преобразований.

Задание 1. анализ обмена данными приложения, с помощью различных анализаторов трафика;

Задание 2. использование режима отладки для поиска нужных участков кода и просмотра данных с которыми работает приложение;

Задание 3 дизассемблирование машинного кода программы (изучение требует довольно много времени);

Задание 4 декомпиляция кода программы для создания исходного кода программы на языке программирования высокого уровня.

ЛИТЕРАТУРА

Основная учебная литература:

Казарин О. В. Программно-аппаратные средства защиты информации. Защита программного обеспечения : учебник и практикум для среднего профессионального образования / О. В. Казарин, А. С. Забабурин. — Москва : Издательство Юрайт, 2021. — 312 с. — (Профессиональное образование). — ISBN 978-5-534-13221-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/476997

Дополнительная учебная литература:

Системы управления технологическими процессами и информационные технологии : учебное пособие для среднего профессионального образования / В. В. Троценко, В. К. Федоров, А. И. Забудский, В. В. Комендантов. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2021. — 136 с. — (Профессиональное образование). — ISBN 978-5-534-09939-3. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/473093