

## ЛАБОРАТОРНАЯ РАБОТА №3

### **Цель работы:**

приобрести умения и практические навыки для работы с командами работы со стеком, командой исполнительного адреса и командами ввода-вывода с использованием функции прерывания *int 21h*.

### **Теоретическая часть:**

#### **Команды работы со стеком**

– *push источник* – поместить данные в стек – размер источника – слово или двойное слово, источником может быть регистр общего назначения, сегментный регистр, ячейка памяти или непосредственный операнд. Регистр *esp (sp)* уменьшается на размер источника, затем выполняется запись в память *ss:[esp]*. Например, *push cx, push 100, push mas;*

– *pop приемник* – считать данные из стека – размер приемника аналогичен предыдущей команде, приемником может быть регистр общего назначения, сегментный регистр (кроме *cs*) или ячейка памяти. Содержимое ячейки памяти по адресу *ss:esp (sp)* копируется в приемник, а затем содержимое *esp (sp)* увеличивается на размер операнда.

#### **Команды тестирования**

– *lea приемник, источник* – команда загрузки (вычисления) исполняемого адреса – приемник в данном случае может быть регистром, источник – переменная, адрес которой задается сложным способом индексации. Если приемник является 16-разрядным, адрес – 32-разрядным, то отбрасываются старшие разряды. Если наоборот – полученное значение дополняется нулями слева. Команду можно использовать для быстрого сложения трех значений. Например, по команде *lea dx, [bx + di]* в регистр *dx* запишется сумма регистров, указанных в квадратных скобках.

## Организация ввода-вывода при помощи функции прерывания *int 21h*:

Команда *int 21h* – команда, с помощью которой программа, написанная на языке ассемблера, запрашивает сервис *DOS* для выполнения определённых действий типа ввода данных с клавиатуры или их вывод на экран. Программа, запрашивающая сервис *DOS*, должна подготовить всю необходимую информацию в регистрах и управляющих блоках, указать в регистре *ah* номер желаемой функции *DOS* и затем вызвать прерывание *int 21h*.

Чтение символа с клавиатуры:

– ввод символа с клавиатуры с эхом, ожиданием и проверкой на Ctrl-Break – *mov ah, 01h* – на выходе регистр *al* содержит код введенного символа;

Вывод символа:

– вывод символа с проверкой на Ctrl-Break – *mov ah, 02h* – на входе *ah* = *02h*, *dl* содержит код выводимого символа;

```
mov ah, 02h
mov dl, 'M'
int 21h    ; на экран будет выведен символ 'M'
```

Чтение строки символов – *mov ah, 0Ah* – *ah* = *0Ah* (10), *ds:dx* – адрес входной строки или буфера.

Первый байт буфера содержит максимально допустимое количество символов при вводе *max*. Остальные байты перед вызовом прерывания содержат подсказку – так называемый «шаблон».

При вводе во второй байт буфера заносится количество введенных символов (не более *max-1*). Вводимые символы размещаются, начиная с третьего байта. Последний символ в буфере – всегда символ *CR* (с кодом *0dh*), который не учитывается в байте длины.

Таким образом, если предполагается ввод строки длиной не более 10 символов, то *max* принимаем равным 11, а под буфер выделяем 12 байтов.

Пример:

```
mov ah, 0ah
lea dx, text; занесли в dx адрес буфера
int 21h     ; в строку text будет введен текст длиной не
более 10 символов
```

```

...
Text db 10, 'место ввода' ; выделяем память под буфер
После ввода строки 'ABC' буфер примет вид:
Text db 10, 3, 'ABC', 0DH, 'о ввода'

```

Вывод строки символов – *mov ah, 09h – ds:dx* содержит адрес выводимой строки, заканчивающейся знаком \$ (этот знак не выводится), регистр *al* содержит код последнего выведенного символа.

```

mov ah, 09h
lea dx, t1      ; адрес начала строки
int 21h         ; будет выведено сообщение 'ravno'
t1 db 'ravno$'

```

**Задание 1.** Ввести с клавиатуры целое число. Выполнить определенные операции с числом в соответствии с выданным вариантом. Ввод и вывод числа выполнять посимвольно. Для формирования двоичного числа из введенных десятичных цифр, воспользоваться блок-схемой, приведенной на рис. 1. Для вывода полученного числа на экран воспользоваться блок-схемой, приведенной на рис. 2.

1. увеличить число в 2 раза и вывести результат на экран.
2. уменьшить число в 2 раза и вывести результат на экран. Если число делится с остатком, то остаток также желательно вывести на экран.
3. увеличить число в 5 раз и вывести результат на экран.
4. уменьшить число в 5 раз и вывести результат на экран. Если число делится с остатком, то остаток также желательно вывести на экран.
5. увеличить число в 10 раз и вывести результат на экран.
6. уменьшить число в 10 раз и вывести результат на экран. Если число делится с остатком, то остаток также желательно вывести на экран.
7. ввести с клавиатуры два числа, сравнить их и вывести на экран наибольшее из них.
8. ввести с клавиатуры два числа, сравнить их и вывести на экран наименьшее из них.

**Задание 2.** Ввести с клавиатуры свою фамилию, вставить ее в строку следующего вида в соответствии с выданным вариантом:

1. «Студент <фамилия> выполнил лабораторную работу №3» и вывести на экран полученную строку.

2. «Студент <фамилия> научился выводить на экран» и вывести на экран полученную строку.

3. «Студент <фамилия> все сделал верно!» и вывести на экран полученную строку.

4. «Студент <фамилия> сдаст системное программирование на отлично» и вывести на экран полученную строку.

5. «Студент <фамилия> умеет программировать на ассемблере» и вывести на экран полученную строку.

6. «Студент <фамилия> может все сдать» и вывести на экран полученную строку.

7. «Студент <фамилия> сдаст все лабораторные» и вывести на экран полученную строку.

8. «Студент <фамилия> знает ассемблер» и вывести на экран полученную строку.

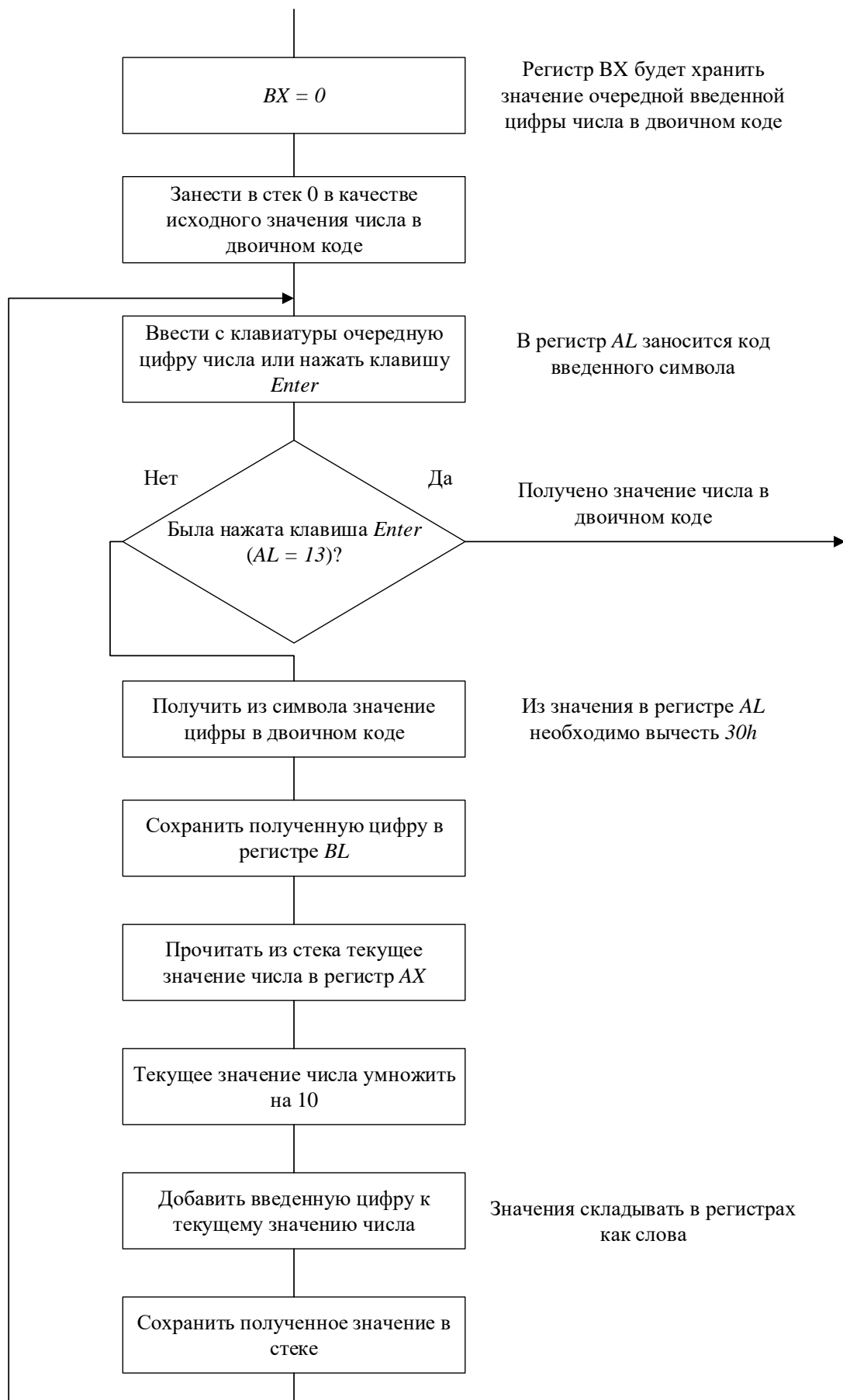


Рис. 1

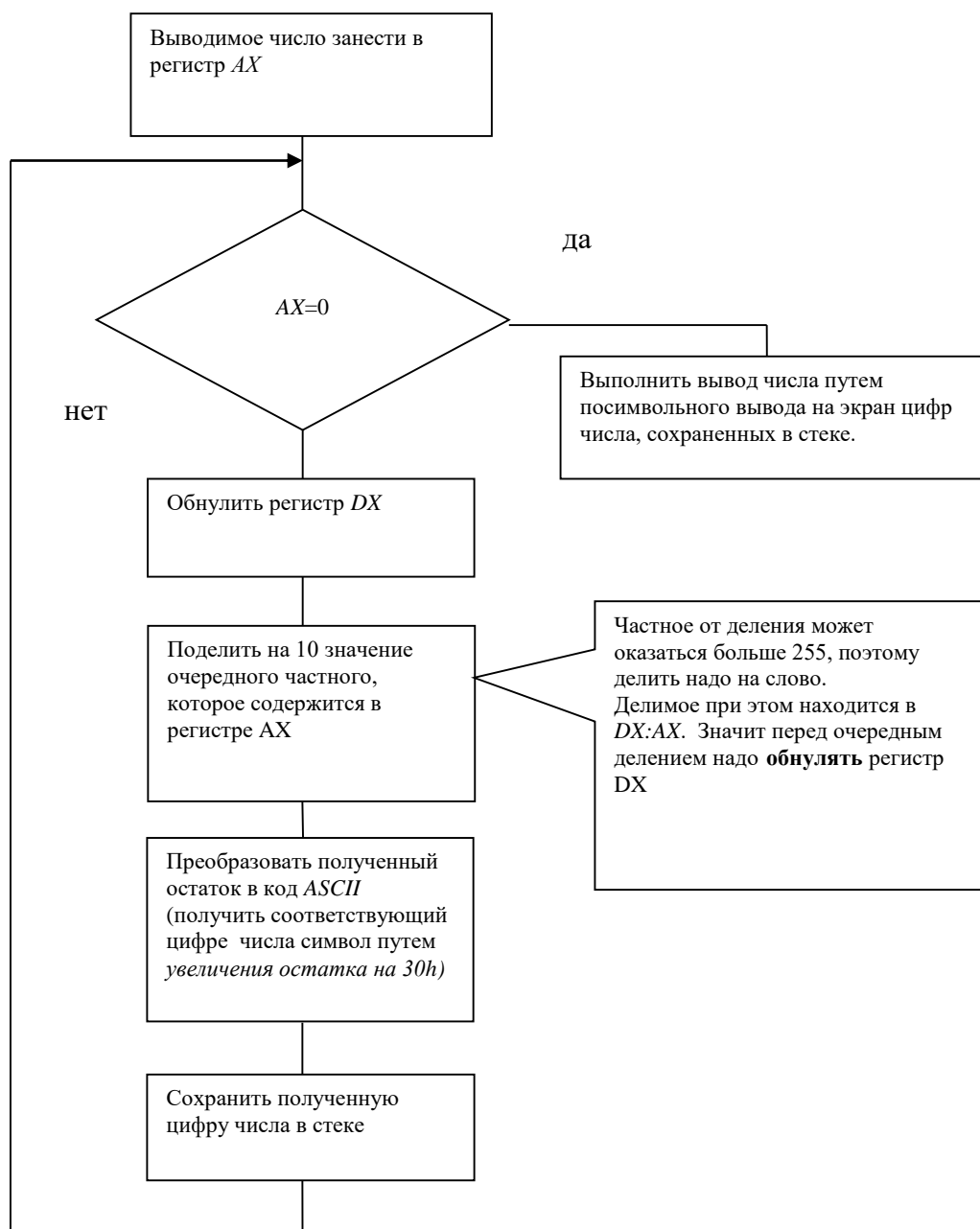


Рис. 2

### ***Порядок работы:***

1. Открыть программу, указанную преподавателем, после объяснения принципов работы с ней.

2. Создать файл с расширением, указанным преподавателем.

3. Ввести текст программы (с крайней левой позиции):

```
.model tiny
.code
N:  push cs
    pop ds
    ; ввести команды программы
    mov ax, 4c00h
    int 21h
.data
    ; ввести директивы распределения данных
end N
```

4. Сохранить программу.

5. Выполнить компиляцию.

6. Если ошибок нет, то запустить эмуляцию программы и пошагово выполнить ее.

7. Подготовить отчет о проделанной работе.

### **Вопросы к теоретическому материалу**

1. Перечислите регистры общего назначения и их названия.
2. Охарактеризуйте регистр *ax*.
3. Охарактеризуйте регистр *bx*.
4. Охарактеризуйте регистр *sx*.
5. Охарактеризуйте регистр *dx*.
6. Опишите строение регистров общего назначения.
7. Что указывает при написании программы на место, где должны располагаться данные?
8. Перечислите псевдокоманды определения данных.
9. Что может быть указано в поле операндов для псевдокоманд определения данных?
10. Какая директива определения данных используется при определении символов и строк?
11. Верна ли следующая запись: *x db 'a'* ? Почему?
12. Верна ли следующая запись: *x dw 'abc'* ? Почему?
13. Каким образом должна быть описана строка, которую необходимо будет вывести на экран?
14. Каким образом должна быть описана строка, в которую планируется сделать запись?
15. Укажите синтаксис команды пересылки данных.
16. Что может быть приемником для команды пересылки данных?
17. Что может быть источником для команды пересылки данных?
18. Укажите синтаксис команды умножения.
19. Укажите синтаксис команды деления.
20. Что является приемником для команд умножения и деления?
21. Что может быть источником для команд умножения и деления?
22. В какой регистр при целочисленном делении записывается остаток от деления?
23. Укажите синтаксис команды для записи значения в стек.



24. Что может быть источником для команды для записи значения в стек?
25. Укажите синтаксис команды для чтения значения из стека.
26. Что может быть приемником для команды чтения значения из стека?
27. Функции какого прерывания используются в данной лабораторной работе для ввода и вывода одиночных символов и строк на экран?
28. В какой регистр записывается номер функции, которую необходимо вызвать?
29. Укажите номер функции для ввода одиночного символа с клавиатуры.
30. В какой регистр запишется введенный с клавиатуры символ?
31. Что именно запишется в качестве символа при его вводе с клавиатуры в регистр *al*?
32. Укажите номер функции для вывода одиночного символа на экран.
33. В какой регистр необходимо занести символ, который должен быть выведен на экран?
34. Укажите номер функции для ввода строки с клавиатуры.
35. В какой регистр и при помощи какой команды необходимо загрузить адрес строки, в которую планируется производить запись?
36. Укажите номер функции для вывода строки на экран.
37. В какой регистр и при помощи какой команды необходимо загрузить адрес строки, которую планируется выводить на экран?

## **ПРОЦЕСС СДАЧИ ЛАБОРАТОРНОЙ РАБОТЫ**

По итогам выполнения каждой лабораторной работы студент:

1. демонстрирует преподавателю правильно работающие программы;
2. демонстрирует приобретенные знания и навыки, отвечая на несколько небольших вопросов преподавателя по составленной программе, возможностям ее доработки и теме лабораторной работы в целом;
3. демонстрирует отчет по выполненной лабораторной работе.

Итоговая оценка складывается из оценок по трем указанным составляющим.

Отчет по лабораторной работе оформляется по шаблону, представленному в приложении 1. Требования к отчету представлены в приложении 2.

**ПРИЛОЖЕНИЕ 1. ШАБЛОН ОТЧЕТА**  
**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ**  
**ФЕДЕРАЦИИ**

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Казанский национальный исследовательский технический университет  
им. А.Н. Туполева – КАИ»

Институт компьютерных технологий и защиты информации  
Отделение СПО ИКТЗИ (Колледж информационных технологий)

ЛАБОРАТОРНАЯ РАБОТА №\_\_  
по дисциплине  
СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ

Работу выполнил

Студент гр.43\_\_

Фамилия И.О.

Принял

Преподаватель Григорьева В.В.

Казань 2024

## **ВАРИАНТ \_\_**

### **1. Цель работы**

**2. Задание на лабораторную работу** – вставляется задание на лабораторную работу, соответствующее индивидуальному варианту студента.

**3. Результат выполнения работы** – формируется описание хода выполнения работы (разработанных подпрограмм, классов, переменных, структур данных и т.п.) и вставляются скриншоты с результатами работы разработанных программ (скриншоты должны быть подписаны).

**4. Листинг программы** – вставляется код разработанной программы с комментариями.

## **ПРИЛОЖЕНИЕ 2. ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ ОТЧЕТА**

Лист документа должен иметь книжную ориентацию, поля документа должны составлять: левое – 3 см, правое – 1,5 см, верхнее – 2 см, нижнее 2 см.

Нумерация страниц – внизу страницы, по центру, особый колонтитул для первой страницы.

Междустрочный интервал – 1,5 (полуторный), отступ первой строки – 1,25.

Текст документа должен быть выполнен с использованием шрифта Times New Roman, размер – 14, выравнивание – по ширине. Заголовки выполняются тем же шрифтом, но размера 16, полужирное начертание, размещение – по центру.

Рисунки должны размещаться по центру, они нумеруются по порядку. Перед рисунком в тексте на него должна быть ссылка. Подпись рисунка должна располагаться по центру и быть выполнена шрифтом Times New Roman, размер – 12. Сначала происходит нумерация рисунка, а затем пишется его название.