Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет

информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра Информатики

Дисциплина «Конструирование программ»

**ОТЧЕТ**

к лабораторной работе №3

на тему:

**«СОЗДАНИЕ ПРОСТОЙ ПРОГРАММЫ НА ЯЗЫКЕ АССЕМБЛЕРА. ОБРАБОТКА СИМВОЛЬНЫХ ДАННЫХ»**

БГУИР 6-05-0612-02 67

|  |
| --- |
| Выполнил студент группы 353503  КОХАН Артём Игоревич |
|  |
| (дата, подпись студента) |
| Проверил ассистент каф. Информатики  РОМАНЮК Максим Валерьевич |
|  |
| (дата, подпись преподавателя) |

Минск 2024

# 1 Индивидуальное задание

**Задание 1. Вариант 3.** Написать программу для вывода на экран нескольких строк символов. Создать ассемблерные программы для двух вариантов исполняемых модулей (com и exe).

**Задание 2. Вариант 16.** Удалить в строке слова, являющиеся числами (учесть разные форматы).

# 2 Выполнение работы

Для настолько простой задачи больше подходит com формат файла, который не хранит дополнительной информации о программе и может использовать только один сегмент памяти. Помимо этого, программу можно упростить для чтения, если добавить макросы, которые реализуют вывод строк.

Листинг кода 1 – Текст программы задания 1 формата com

.model tiny

.data

message1 db "qwertyuiop[]\asdfghjkl;'zxcvbnm,./1234567890!@#`~:<>#@|+=\_-",0Dh,0Ah,'$'

message2 db "ASSembler",0Dh,0Ah,'$'

message3 db "It's very hard program",0Dh,0Ah,'$'

.code

print macro out\_str

mov AH,9

mov DX,offset out\_str

int 21h

endm

start:

print message1

print message2

print message3

end start

Листинг кода 2 – Текст программы задания 1 формата exe

.model small

.stack 100h

.data

message1 db "qwertyuiop[]\asdfghjkl;'zxcvbnm,./1234567890!@#`~:<>#@|+=\_-",0Dh,0Ah,'$'

message2 db "ASSembler",0Dh,0Ah,'$'

message3 db "It's very hard program",0Dh,0Ah,'$'

.code

print macro out\_str

mov AH, 9

lea DX, out\_str

int 21h

endm

main proc

mov AX, @data

mov DS, AX

print message1

print message2

print message3

mov AX, 4C00h

int 21h

main endp

end main

Согласно условию необходимо учесть разные форматы чисел, программа удаляет из строки целые числа, числа с плавающей точкой, а также, экспонентная запись числа. При том что программа обрабатывает при записи чисел с плавающей точкой, как знак запятой, так и знак точки. А при записи в экспонентной форме числа обрабатывает, как большую ‘Е’, так и маленькую ‘е’. В конечной итоге, сопровождая комментариями, результат выводится через пробел.

Листинг кода 3 – Текст программы задания 2

.model small;

.stack 100h;

.data

CountPoints db 0h

CountExp db 0h

Info db 'Program removes integers, floating point numbers with , or . and exp numbers! $'

InputStr db 'Input string: $'

OutputStr db 'Output string: $'

Str db 1000h dup(0h)

NewStr db 1000h dup(0h)

.code

print macro out\_str

mov AH,9

mov DX,offset out\_str

int 21h

endm

newline macro

mov AH, 2

mov DL, 13

int 21h

mov DL, 10

int 21h

endm

Start:

MOV AX, @data

MOV DS, AX

print Info

newLine

print InputStr

MOV AH, 1h

MOV SI, 0h

MOV BX, 0h

Input:

INT 21h

CMP AL, 13

JE EndInput

MOV Str[BX+SI], AL

INC SI

CMP AL, ' '

JNE Input

ADD BX, 50h

MOV SI, 0h

JMP Input

EndInput:

MOV Str[BX+SI], 0h

MOV BX, 0h

MOV SI, 0h

MOV DX, 0h

CheckOnlyNumInWord:

MOV AL, Str[BX+SI]

CMP AL, 0h

JE LastWord

CMP AL, ' '

JE DoneWord

CMP AL, '.'

JE IsPoint

CMP AL, ','

JE IsPoint

CMP AL, 'e'

JE IsExp

CMP AL, 'E'

JE IsExp

CMP AL, '0'

JB NotNumber

CMP AL, '9'

JA NotNumber

INC SI

JMP CheckOnlyNumInWord

NotNumber:

MOV CX, 1h ; need to new

INC SI

JMP CheckOnlyNumInWord

NumberDot:

CMP CountPoints, 1h

JA NotNumber

MOV AL, Str[BX+SI+1]

CMP AL, 0h

JE NotNumber

CMP AL, ' '

JE NotNumber

INC SI

JMP CheckOnlyNumInWord

NumberExp:

CMP CountExp, 1h

JA NotNumber

MOV AL, Str[BX+SI+1]

CMP AL, 0h

JE NotNumber

CMP AL, ' '

JE NotNumber

INC SI

JMP CheckOnlyNumInWord

IsPoint:

MOV AL, CountPoints

ADD AL, 1h

MOV CountPoints, AL

CMP SI, 0h

JA NumberDot

JMP NotNumber

IsExp:

MOV AL, CountExp

ADD AL, 1h

MOV CountExp, AL

CMP SI, 0h

JA NumberExp

JMP NotNumber

DoneWord:

MOV DI, 1h

MOV CountPoints, 0h

MOV CountExp, 0h

AfterToNewStr:

CMP CX, 1h

JE ToNewStr

ADD BX, 50h

CMP DI, 0h

JE AddDX

MOV SI, 0h

MOV CX, 0h

JMP CheckOnlyNumInWord

AddDX:

ADD DX, 50h

MOV CX, 0h

MOV SI, 0h

JMP CheckOnlyNumInWord

ToNewStr:

MOV CX, SI

MOV DI, 0h

RepeatAdd:

ADD BX,CX

SUB BX, SI

MOV AL, Str[BX] ; BX+CX-SI

ADD BX, SI

SUB BX,CX

CMP AL, 0h

JE LastWord

ADD DI, DX

ADD DI, CX

SUB DI, SI

MOV NewStr[DI], AL ; DX+CX-SI

ADD DI, SI

SUB DI, DX

SUB DI, CX

CMP SI, 0h

JE AfterToNewStr

DEC SI

JMP RepeatAdd

LastWord:

CMP CX, 1h

JE ToNewStr

MOV AH, 2h

MOV BH, 0h

MOV DH, 2h

MOV DL, 0h

INT 10h

MOV BX, 0h

MOV SI, 0h

MOV AH, 9h

MOV DX, offset OutputStr

INT 21h

MOV AH, 2h

Output:

ADD BX, SI

MOV AL, NewStr[BX]

SUB BX, SI

MOV DH, 0h

MOV DL, AL

INC SI

CMP AL, 0h

JE ENDProgram

CMP AL, ' '

JNE OutS

MOV SI, 0h

ADD BX, 50h

MOV AL, ' '

OutS:

INT 21h

JMP Output

ENDProgram:

END start

# Вывод

В ходе выполнения лабораторной работы с программным обеспечением, предназначенным для сборки, отладки и запуска программ на языке ассемблера ознакомлен с основными особенностями архитектуры процессора и общей структурой программы. Получены навыки с директивами определения данных, командами пересылки данных и передачи управления, строчные операции и прерывания консольного ввода-вывода высокого уровня.